

AD-A051 519

AIR FORCE AVIONICS LAB WRIGHT-PATTERSON AFB OHIO  
GEANS SOFTWARE ANALYSIS DEVELOPMENT AND VALIDATION.(U)  
AUG 77 W MIKULSKI, J E BARNES

F/G 17/7

UNCLASSIFIED

AFAL-TR-77-119

NL

1 OF 1  
AD  
A051519



AD A051519

AFAL-TR-77-119

2

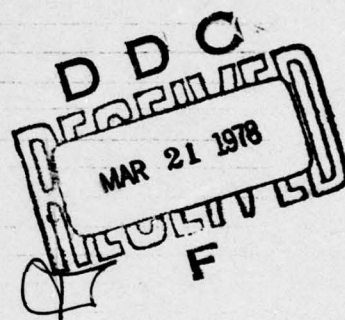


## GEANS SOFTWARE ANALYSIS DEVELOPMENT AND VALIDATION

Reference Systems Branch  
Reconnaissance and Weapon Delivery Division

August 1977

TECHNICAL REPORT AFAL-TR-77-119  
FINAL REPORT FOR PERIOD NOVEMBER 1975 TO NOVEMBER 1976



Approved for public release; distribution unlimited.

AIR FORCE AVIONICS LABORATORY  
AIR FORCE WRIGHT AERONAUTICAL LABORATORIES  
AIR FORCE SYSTEMS COMMAND  
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433

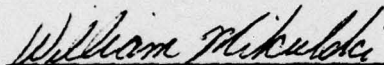
AD No. \_\_\_\_\_  
DDC FILE COPY

NOTICE

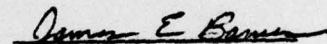
When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.


This report has been reviewed by the Information Office (OI) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

  
WILLIAM MIKULSKI  
Project Engineer

FOR THE COMMANDER

  
JAMES E. BARNES  
Project Engineer

  
CHARLES L. HUDSON, Colonel, USAF  
Chief  
Reconnaissance and Weapon Delivery Division

"If your address has changed, if you wish to be removed from our mailing list, or if the addressee is no longer employed by your organization please notify AFAL/RWA, W-PAFB, OH 45433 to help us maintain a current mailing list".

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.



SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER <b>AFAL-TR-77-119</b>	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) <b>GEANS SOFTWARE ANALYSIS DEVELOPMENT AND VALIDATION</b>	5. TYPE OF REPORT & PERIOD COVERED <b>Final Report Nov 75 - Nov 76</b>	
7. AUTHOR <b>William Mikulski James E. Barnes</b>	8. CONTRACT OR GRANT NUMBER(s)	
9. PERFORMING ORGANIZATION NAME AND ADDRESS <b>Air Force Avionics Laboratory (AFAL/RWA-3) Air Force Wright Aeronautical Laboratories AFSC, Wright-Patterson AFB, Ohio 45433</b>	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS <b>Project 19278203</b>	
11. CONTROLLING OFFICE NAME AND ADDRESS <b>Same as block 9 above</b>	12. REPORT DATE <b>Aug 77</b>	13. NUMBER OF PAGES <b>79</b>
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report) <b>UNCLASSIFIED</b>	
15a. DECLASSIFICATION/DOWNGRADING SCHEDULE		
16. DISTRIBUTION STATEMENT (of this Report)  <b>Approved for public release; distribution unlimited.</b>		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)  <b>Computer program High order language Assembly language</b>		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number)  <b>The GEANS Software Analysis Development and Validation effort consisted of writing an existing assembly language program in a high order language (HOL) and then comparing the two-load modules for memory requirements and timing. This comparison showed that the program as written in the HOL required more memory and time than the same program as written in assembly language.</b>		

**DDC**  
**RECEIVED**  
**MAR 21 1978**  
**F**

017 670



SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

REPORT DOCUMENTATION PAGE	
1. REPORT NUMBER	2. GOVT ACCESSION NO.
3. AUTHOR(s)	4. TITLE
5. AUTHORING ORGANIZATION NAME(S) AND ADDRESS(ES)	6. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
7. AUTHORING ORGANIZATION REPORT NUMBER	8. PERFORMING ORGANIZATION REPORT NUMBER
9. PROGRAM ELEMENT NAME(S)	10. PROGRAM ELEMENT NUMBER
11. PROGRAM ELEMENT NUMBER	12. PROGRAM ELEMENT NUMBER
13. PROGRAM ELEMENT NUMBER	14. PROGRAM ELEMENT NUMBER
15. PROGRAM ELEMENT NUMBER	16. PROGRAM ELEMENT NUMBER
17. PROGRAM ELEMENT NUMBER	18. PROGRAM ELEMENT NUMBER
19. PROGRAM ELEMENT NUMBER	20. PROGRAM ELEMENT NUMBER
21. PROGRAM ELEMENT NUMBER	22. PROGRAM ELEMENT NUMBER
23. PROGRAM ELEMENT NUMBER	24. PROGRAM ELEMENT NUMBER
25. PROGRAM ELEMENT NUMBER	26. PROGRAM ELEMENT NUMBER
27. PROGRAM ELEMENT NUMBER	28. PROGRAM ELEMENT NUMBER
29. PROGRAM ELEMENT NUMBER	30. PROGRAM ELEMENT NUMBER
31. PROGRAM ELEMENT NUMBER	32. PROGRAM ELEMENT NUMBER
33. PROGRAM ELEMENT NUMBER	34. PROGRAM ELEMENT NUMBER
35. PROGRAM ELEMENT NUMBER	36. PROGRAM ELEMENT NUMBER
37. PROGRAM ELEMENT NUMBER	38. PROGRAM ELEMENT NUMBER
39. PROGRAM ELEMENT NUMBER	40. PROGRAM ELEMENT NUMBER
41. PROGRAM ELEMENT NUMBER	42. PROGRAM ELEMENT NUMBER
43. PROGRAM ELEMENT NUMBER	44. PROGRAM ELEMENT NUMBER
45. PROGRAM ELEMENT NUMBER	46. PROGRAM ELEMENT NUMBER
47. PROGRAM ELEMENT NUMBER	48. PROGRAM ELEMENT NUMBER
49. PROGRAM ELEMENT NUMBER	50. PROGRAM ELEMENT NUMBER
51. PROGRAM ELEMENT NUMBER	52. PROGRAM ELEMENT NUMBER
53. PROGRAM ELEMENT NUMBER	54. PROGRAM ELEMENT NUMBER
55. PROGRAM ELEMENT NUMBER	56. PROGRAM ELEMENT NUMBER
57. PROGRAM ELEMENT NUMBER	58. PROGRAM ELEMENT NUMBER
59. PROGRAM ELEMENT NUMBER	60. PROGRAM ELEMENT NUMBER
61. PROGRAM ELEMENT NUMBER	62. PROGRAM ELEMENT NUMBER
63. PROGRAM ELEMENT NUMBER	64. PROGRAM ELEMENT NUMBER
65. PROGRAM ELEMENT NUMBER	66. PROGRAM ELEMENT NUMBER
67. PROGRAM ELEMENT NUMBER	68. PROGRAM ELEMENT NUMBER
69. PROGRAM ELEMENT NUMBER	70. PROGRAM ELEMENT NUMBER
71. PROGRAM ELEMENT NUMBER	72. PROGRAM ELEMENT NUMBER
73. PROGRAM ELEMENT NUMBER	74. PROGRAM ELEMENT NUMBER
75. PROGRAM ELEMENT NUMBER	76. PROGRAM ELEMENT NUMBER
77. PROGRAM ELEMENT NUMBER	78. PROGRAM ELEMENT NUMBER
79. PROGRAM ELEMENT NUMBER	80. PROGRAM ELEMENT NUMBER
81. PROGRAM ELEMENT NUMBER	82. PROGRAM ELEMENT NUMBER
83. PROGRAM ELEMENT NUMBER	84. PROGRAM ELEMENT NUMBER
85. PROGRAM ELEMENT NUMBER	86. PROGRAM ELEMENT NUMBER
87. PROGRAM ELEMENT NUMBER	88. PROGRAM ELEMENT NUMBER
89. PROGRAM ELEMENT NUMBER	90. PROGRAM ELEMENT NUMBER
91. PROGRAM ELEMENT NUMBER	92. PROGRAM ELEMENT NUMBER
93. PROGRAM ELEMENT NUMBER	94. PROGRAM ELEMENT NUMBER
95. PROGRAM ELEMENT NUMBER	96. PROGRAM ELEMENT NUMBER
97. PROGRAM ELEMENT NUMBER	98. PROGRAM ELEMENT NUMBER
99. PROGRAM ELEMENT NUMBER	100. PROGRAM ELEMENT NUMBER

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

Blank

AFAL-TR-77-119

# FOREWORD

This report was prepared by William Mikulski and James E. Barnes of the Reference Systems Branch, Reconnaissance and Weapon Delivery Division, Air Force Avionics Laboratory, Wright-Patterson Air Force Base, Ohio.

The work was initiated under Project Work Unit Number 19270203. The report covers effort during the period November 1975 to November 1976.

ACCESSION for	
NTIS	White Section <input checked="" type="checkbox"/>
DDC	Buff Section <input type="checkbox"/>
UNANNOUNCED	<input type="checkbox"/>
JUSTIFICATION	<input type="checkbox"/>
BY	
DISTRIBUTION/AVAILABILITY CODES	
Dist.	SPECIAL
A	

TABLE OF CONTENTS

SECTION	PAGE
I INTRODUCTION	1
II BACKGROUND OF JOVIAL/J3B	2
III INTEGRATION OF HOL WITH EXISTING ASSEMBLY LANGUAGE PROGRAM	4
1. Approach	4
2. Problems Encountered	5
IV MEMORY AND TIMING REQUIREMENTS	7
V DISCUSSION OF RESULTS	10
VI SUMMARY AND CONCLUSIONS	12
APPENDIX FLOWCHARTS	15



## TABLES

TABLE		PAGE
1	Memory Requirements	7
2	Comparison of GEANS timing in J3B and FOCAP	9

## SECTION I

### INTRODUCTION

In May 1973 AFAL/RWA-3 undertook the task of converting the Honeywell Gimballled Electrostatic Gyro Navigation System (GEANS) computer Program program, which was hosted on the Honeywell HDC-601 computer, to run on the Singer/Kearfott SKC-2000 computer. Both the original and converted programs were written in the assembly language of the host computer. Thus, the conversion effort was an assembly language to assembly language conversion, and was completed in December 1975. This effort is described in Technical Report AFAL-TR 77-8, Vol 1 & 2, Conversion of Computer Software for the Gimballled Electrostatic Gyro Navigation System, November 1976.

GEANS is intended for a variety of applications using any of several available computers. The required reprogramming will be most rapidly done using Higher Order Languages (HOL), if a suitable HOL is available. Good HOL's increase programmer productivity, provide readable programs, and, for some languages, allow the same HOL source program to be used on several computers.

The purpose of the investigation described in this report was to evaluate the problems and penalties of using an avionics-oriented HOL to reprogram the GEANS software. The available time and resources limited the effort to applying a readily available HOL, J3B level 0, to only the Alignment and Navigation portions of the GEANS Software. These portions were converted and the resulting computer time and memory required compared with those of the assembly language code.

SECTION II  
BACKGROUND OF JOVIAL/J3B

The compiler used was the JOVIAL-3B/O compiler written by Softech Inc. and hosted on the IBM-370/155 at the WPAFB computer center. Following is a short history of JOVIAL-3B:

When the Request for Proposal for the B-1 bomber system was prepared at Aeronautical Systems Division (ASD), it was decided to require the programming of the mission software to be done in a HOL. In October 1971 JOVIAL-73 was selected as the language for the B-1. However the JOVIAL-73 language was not ready in time so in June 1972 a reduced capability HOL of low implementation risks which would satisfy the minimum needs of the B-1 until JOVIAL-73 became available was designed. To achieve low implementation risk, this language was a version of JOVIAL-5, which is a subset of JOVIAL-73. The language was designed by Boeing and three members of the JOVIAL-73 committee, namely RADC, ASD (the B-1 SPO), and ABACUS, Inc. Whenever possible the language was made to conform with what was known of JOVIAL-73 at the time.

The contract for a JOVIAL-3B compiler was finally awarded to a contractor (Softech) who did not use the JOVIAL-5 compiler as a baseline. Thus the need for staying close to JOVIAL-5 was eliminated. Since much more was known about the direction of JOVIAL-73 by the time the JOVIAL-3B specification was finished in October 1972 (by Boeing and Softech), it was very similar to JOVIAL-73.



Two compilers for JOVIAL-3B, one for the IBM-360 and one for the SKC-2000 were written in the Automated Engineering Design (AED) language and hosted on the IBM-360. They were delivered to Boeing in September 1973.

The original reduced capability version of JOVIAL-3B has since come to be termed JOVIAL-3B/0 (J3B level 0). It has evolved into JOVIAL-3B/1 and, later, JOVIAL-3B/2. The level 2 compiler is the most current version, and can be targeted to the LC-4516D and M362F computers as well as to the SKC-2000 and IBM-370.

### SECTION III

#### INTEGRATION OF THE HOL GEANS PROGRAM WITH EXISTING ASSEMBLY LANGUAGE PROGRAM

##### 1. APPROACH

Coding of the GEANS software in JOVIAL/J3B was separated into three parts. These were functionally different and consisted of:

- 1) Navigation routines.
- 2) Alignment routines.
- 3) Matrix operation subroutines.

Since these were the only portions coded in J3B, it was necessary to link with the existing software written in the SKC-2000 assembly language (FOCAP). To minimize debugging time and effort, the navigation routines were coded in J3B almost line for line from the hand-coded routines. The alignment routines were written based on the flowcharts rather than on the FOCAP code itself. The FOCAP code was checked with the flowcharts to verify the accuracy of the details only. The matrix routines were written in J3B to minimize linkage problems which would have been encountered if the FOCAP written matrix routines were used.

The same labels were used in the J3B code as were used in the FOCAP code, which allowed for greater ease in both timing and debugging. The same data base was also used since the J3B written programs had to link with the hand-written FOCAP programs. Using the same data base caused some problems since the J3B generated code does not use FOCAP "common." This problem was alleviated by modifying the FOCAP-Coded data base to refer to the J3B defined data blocks.

## 2. PROBLEMS ENCOUNTERED

Since JOVIAL/J3B is target machine independent, it cannot generate input/output code. Because of this, all I/O must be done by handwritten FOCAP routines. The required routines for I/O already existed in the GEANS real-time executive. Therefore, to prevent the additional problem of debugging I/O routines, the J3B code was written to look like the hand-written FOCAP code as far as the executive (hand-written) was concerned. This was done by maintaining the same names for all routines which the executive referenced.

The executive code also had to be modified slightly since J3B generates code using "Page 3", which allows short instructions to be generated which access constants 0 and 1 from core. This required adding code in the executive to set the status word to indicate that a page 3 was in effect.

Another modification to the executive was required for the J3B linkage mechanization. J3B uses a push-down stack arrangement using registers 6 and 14 as stack pointers. The existing GEANS software used register 6 only for the same purpose. Therefore the executive had to be modified to initialize register 14 to the same as register 6 (i.e. to the top of the stack).

The existing math library written for the GEANS in the SKC-2000 had been previously proven to work properly and efficiently. Therefore, it was decided to use this library of math functions rather than the library supplied with the J3B compiler. The only drawback to this



method was that the job linkage and the linkage used by the FOCAP coded library were entirely different. This required additional code to convert from one calling convention and linkage structure to the other. The hand-written math library was written to be most efficient with the hand-written GEANS program. Rather than modify this library, it was decided to keep it intact and add a buffer stage of routines to convert from one linkage convention to the other. The standard J3B linkage convention passes addresses of all arguments in Reg A and Reg B, with all others following the call to the subroutine. The hand-coded library, however, passed the actual argument in the A-B registers in some cases, used index register 4 in other cases and returned the result in the A-B register, or a specified address. This inconsistency in linkage procedures required separate intermediate linkage conversion routines for each routine used in the math library. Standard JOVIAL/J3B linkage convention requires saving index registers 1 thru 5 (XR1-XR5) on the stack, obtaining proper arguments or addresses and placing them at the appropriate location according to the routine being called. Upon return from the subroutine, the results had to be returned to the J3B caller where proscribed by the J3B linkage convention. Also, the index registers (XR1-XR5) had to be restored from the stack and the stack cleaned up to the state it was in when called by the J3B program.

## SECTION IV

## MEMORY AND TIME REQUIREMENTS

Memory requirements for both the J3B and FOCAP programs were taken from the memory map produced by the Linkage Editor step of the compilation or assembly. Table 1 gives a comparison of these requirements.

TABLE 1

## MEMORY REQUIREMENTS

(SKC 2000 32 Bit Words)

For routines converted directly from FOCAP to J3B:

	<u>NAV</u>	<u>ALIGN</u>	<u>INIT</u>	<u>TOTAL</u>
J3B	885	1477	744	3106
FOCAP	1014	836	694	2544

Total memory requirements, including the executive (RTEXEC) and math subroutine library (SUBLIB) plus linkage routines:

FOCAP	4524
J3B	6350

Timing for these programs are listed in Table 2. All major routines were timed using the following method: A Hewlett-Packard 1600S Logic State Analyzer, consisting of a H-P 1600A Logic State Analyzer and a HP 1607 Logic State Analyzer was connected to the address lines of the SKC-2000 at the Computer Control Unit (CCU). The H-P 1600A address compare lines were set to the desired start address of the GEANS program and the H-P 1607 address compare lines were set to the desired stop address. When the address compare lines matched the start address in the SKC-2000 the H-P 1600A generated a trigger which was fed to an electronic counter. The H-P 1607 generated a trigger for the stop address in like manner. The electronic counter recorded the time delay between the start and stop triggers, thus giving the time of execution of the block of code under test. The uncertainty in this method is 100 nanoseconds, the sensitivity of the electronic counter. The accuracy of these measurements is  $\pm 100$  nanoseconds.



TABLE 2

## COMPARISON OF GEANS TIMING IN J3B AND FOCAP

(Subroutine description in flowchart form are in the Appendix.)

<u>SUBROUTINE</u>	<u>TIME IN MICROSECONDS</u>	
	<u>FOCAP</u>	<u>J3B</u>
IIC	1337	3000
IID	2620	4500
IIE	1250	2750
IIF	8417	16975
IIG	3480	12204
IIH	3162	4670
IIK	1183	1756
IIM	1636	2542
IIO	3642	6318
IIP	13066	47345
RSET	347	730
IIR	45	39
FENT	6676	17151
DECD	2340	2440
IE	2460	3900
IF	977	1685
IG	790	1685
IH	3883	5400
IJ	7254	7800

SECTION V  
DISCUSSION OF RESULTS

For that portion of GEANS that was written in J3B (Alignment, Navigation and Initialization) the memory requirement was approximately 20% greater for the J3B version than for the FOCAP version. The total memory requirement for J3B was 29% greater than for FOCAP. The J3B compiler produced approximately 80% short instructions (a short instruction is 1/2 word, or 16 bits, long). The FOCAP version produced approximately 15% short instructions.

The J3B version was found to run two to three times slower than the FOCAP version. A number of things contributed to this:

1) The compiler was designed to create a high density of short instructions. This it does, and actually produces more instructions than necessary as a result. It reserves Index Register 5 (XR5) as a base register and loads XR5 with the address of one of several data areas to create short instructions. As a result when some operations are being performed, such as creation of a 3x3 matrix, the compiler loads XR5 once for each data item that is moved. So an index register load is performed (which takes one full word of memory and 2.5 microseconds) to create two short instructions. This sequence is repeated nine times, and is very inefficient both in time and memory usage.

2) The compiler generates code which computes an address offset each time it is used within a loop. For example, if the offset is used four times within a loop it is computed four times, and it only needs to be computed the first time.

3) The special linkage subroutines, those that resolve the J3B/FOCAP linkage differences, consumed 50 microseconds. Each subroutine call in GEANS required 100 microseconds more to execute because of this requirement.

The JOVIAL-3B level 0 compiler is poorly optimized to save time of execution. It will produce a high density of instructions, and with a data base designed to take advantage of this ability some optimization in time and memory is possible. For this effort the FOCAP data base was used, so the shortcomings of the compiler were accentuated. Later versions may be more efficient.

## SECTION VI

### SUMMARY AND CONCLUSIONS

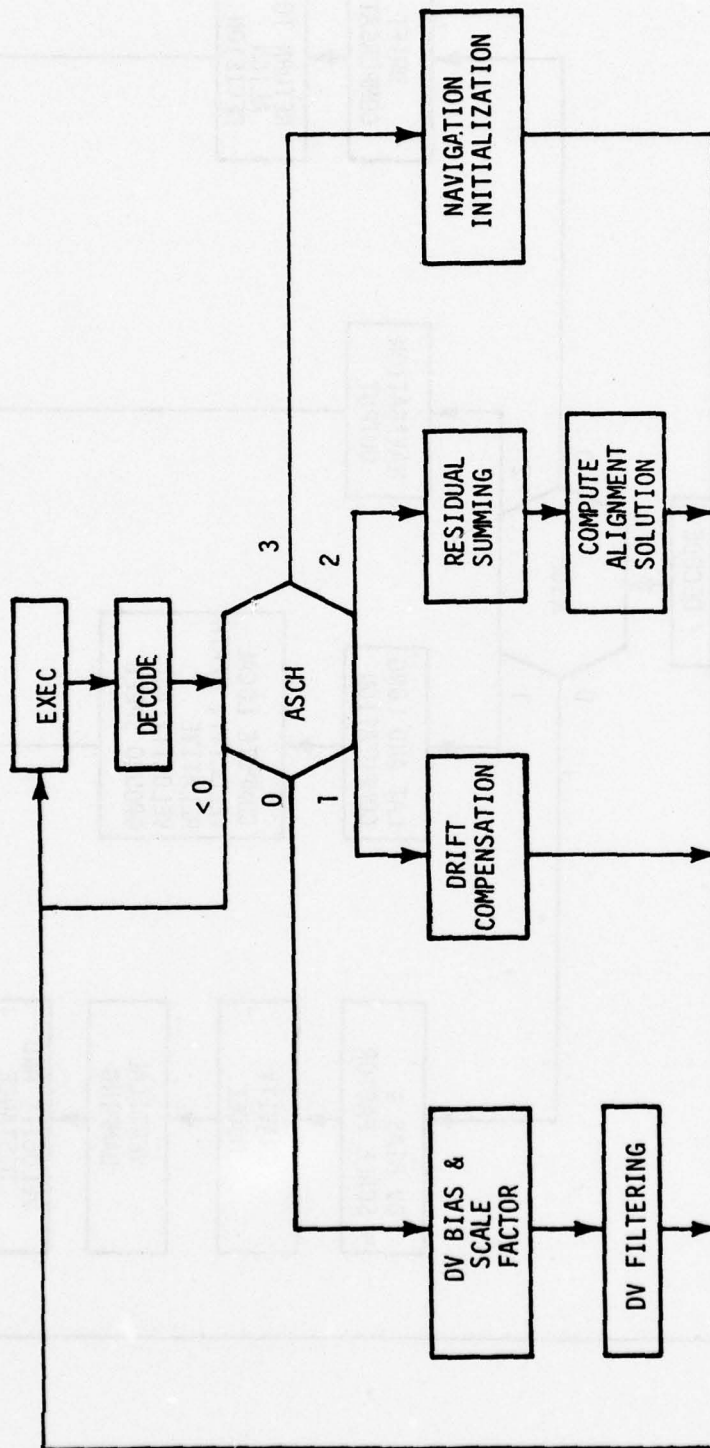
The GEANS software development effort consisted of reprogramming the GEANS Alignment and Navigation algorithms in a High Order Language (HOL), JOVIAL/J3B, Level 0. The GEANS program was hosted on the Singer/Kearfott SKC-2000 computer and was written in the SKC-2000 assembly language, FOCAP. The purpose of the J3B effort was to compare the memory and time requirements of GEANS as written in FOCAP to GEANS written in J3B. Alignment, navigation and initialization portions of GEANS were coded in J3B. The FOCAP data base, math subroutine library, and real time executive were retained as part of the J3B version.

Final results showed that the J3B version required 29% more memory and two to three times more time than the FOCAP version. The memory requirement of 29% is somewhat misleading because additional code had to be written to resolve linkage convention differences between the FOCAP and J3B versions. In those routines that were coded in J3B directly from FOCAP (i.e. Alignment and Navigation) the J3B version required 20% more memory than FOCAP. A similar case can be made for timing requirements. The particular compiler used for this study (J3B Level 0) is a very inefficient compiler and is very poorly optimized. If GEANS had been written from scratch in J3B, with a properly designed data base and real time executive, a considerable increase in timing and memory efficiency might have been realized over the results of this study.

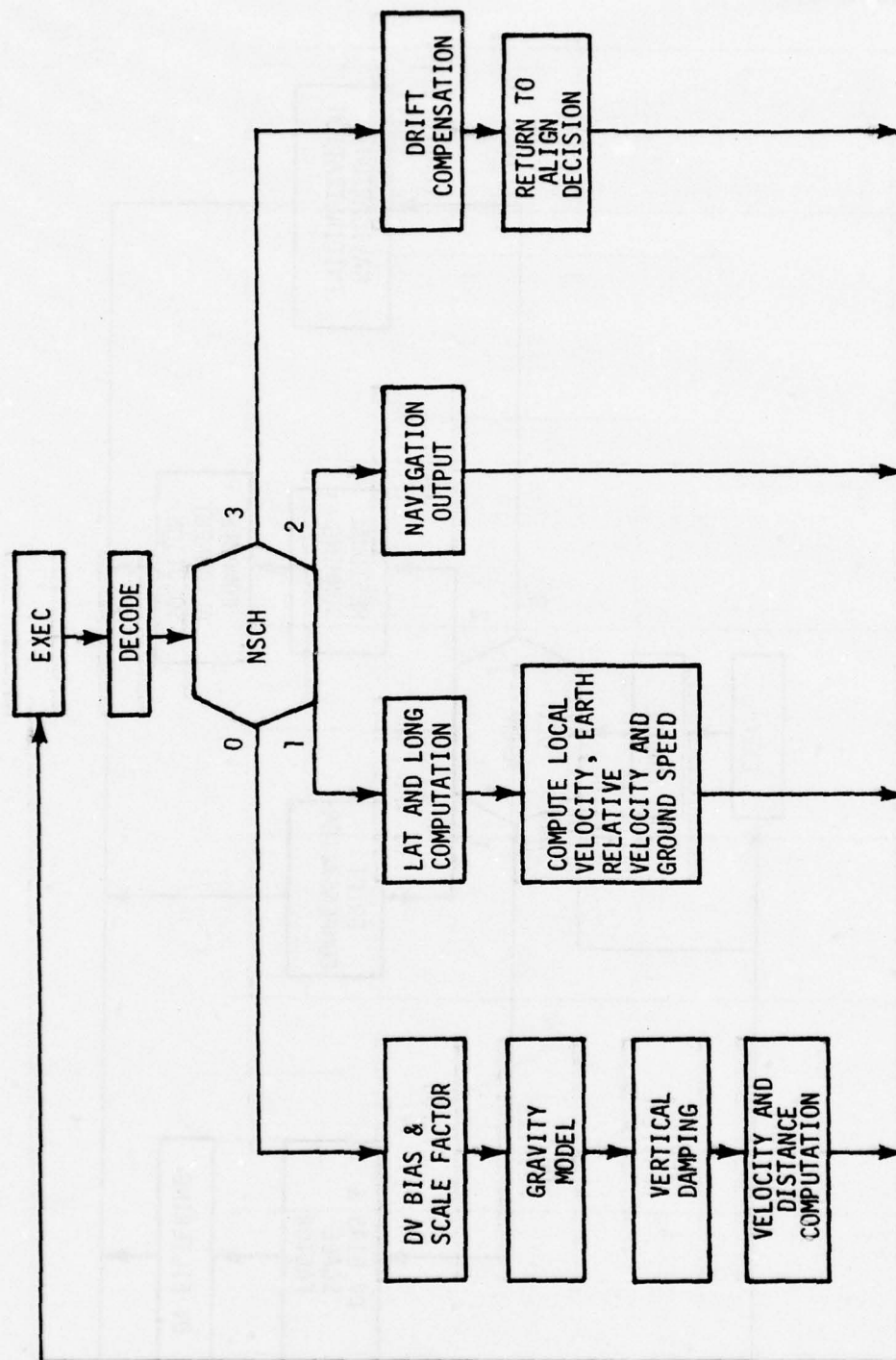


The results of this study were brought to the attention of Softech, Inc. (designers of J3B), who explained that the latest version of J3B (J3B Level 2) corrects most, if not all, of the deficiencies of J3B level 0. A more efficient compiler would most definitely show better results. The overhead for an efficient compiler of any high order language would be close to 20% more than assembly language.

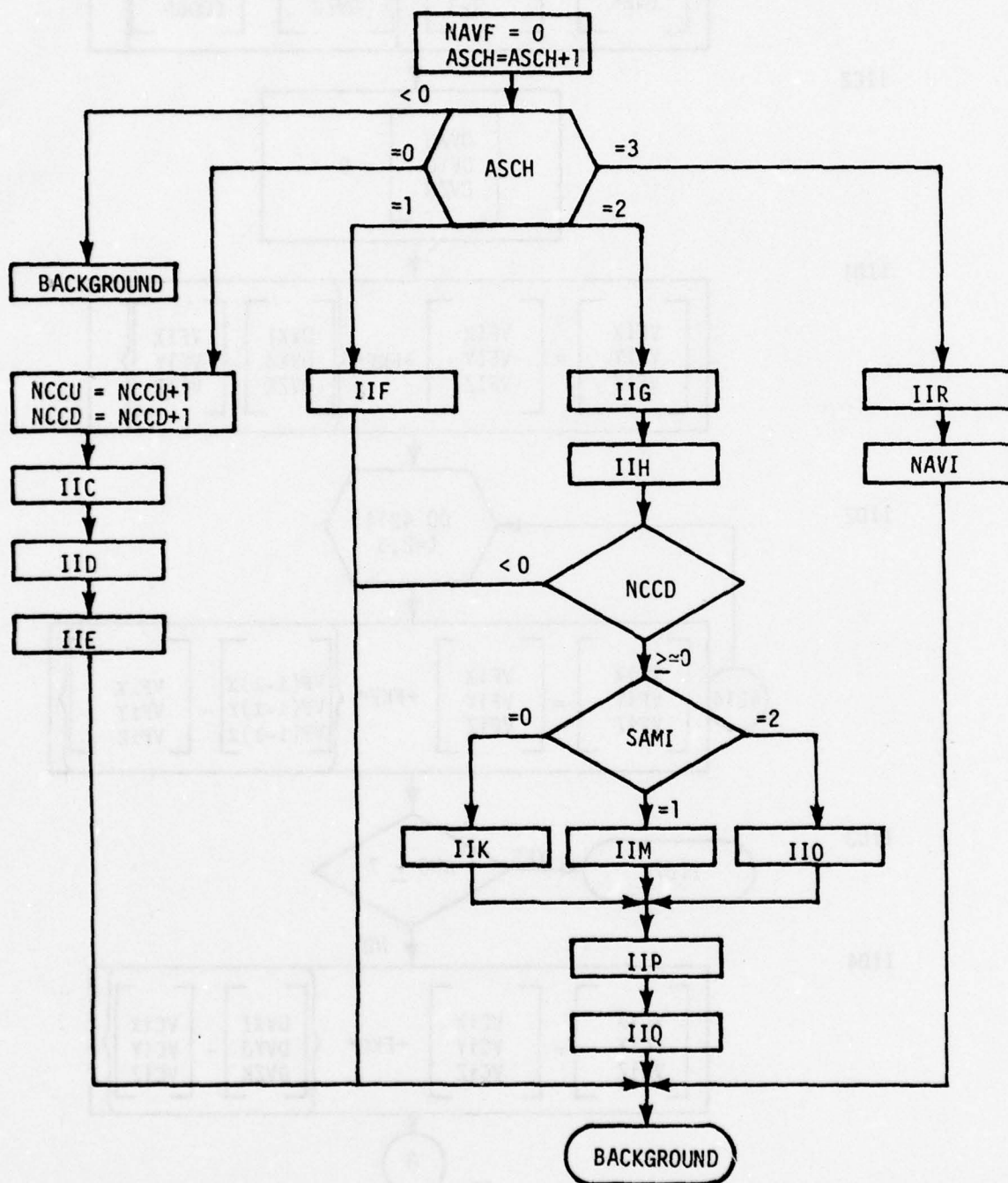
APPENDIX  
FLOWCHARTS  
GEANS ALIGNMENT



GEANS NAVIGATION



## ALIGNMENT SUB-EXECUTIVE





## LOW PASS FILTER

IIC=IC

IIC1

$$\begin{bmatrix} DVXI \\ DVXJ \\ DVZK \end{bmatrix} = \begin{bmatrix} AB \\ 3 \times 3 \end{bmatrix} \begin{bmatrix} DVXG \\ DVG \\ DVZG \end{bmatrix} - \begin{bmatrix} CD04D \\ CD05D \\ CD06D \end{bmatrix}$$

IIC2

$$\begin{bmatrix} DVXG \\ DVG \\ DVZG \end{bmatrix} = 0$$

IID1

$$\begin{bmatrix} VFIX \\ VFIY \\ VFIZ \end{bmatrix} = \begin{bmatrix} VFIX \\ VFIY \\ VFIZ \end{bmatrix} + FKF * \left\{ \begin{bmatrix} DVXI \\ DVG \\ DVZK \end{bmatrix} - \begin{bmatrix} VFIY \\ VFIY \\ VFIZ \end{bmatrix} \right\}$$

IID2

DO 4214  
I=2,3

$$\begin{bmatrix} VFiX \\ VFiY \\ VFiZ \end{bmatrix} = \begin{bmatrix} VFiX \\ VFiY \\ VFiZ \end{bmatrix} + FKF * \left\{ \begin{bmatrix} VF(i-1)X \\ VF(i-1)Y \\ VF(i-1)Z \end{bmatrix} - \begin{bmatrix} VFiY \\ VFiY \\ VFiZ \end{bmatrix} \right\}$$

IID3

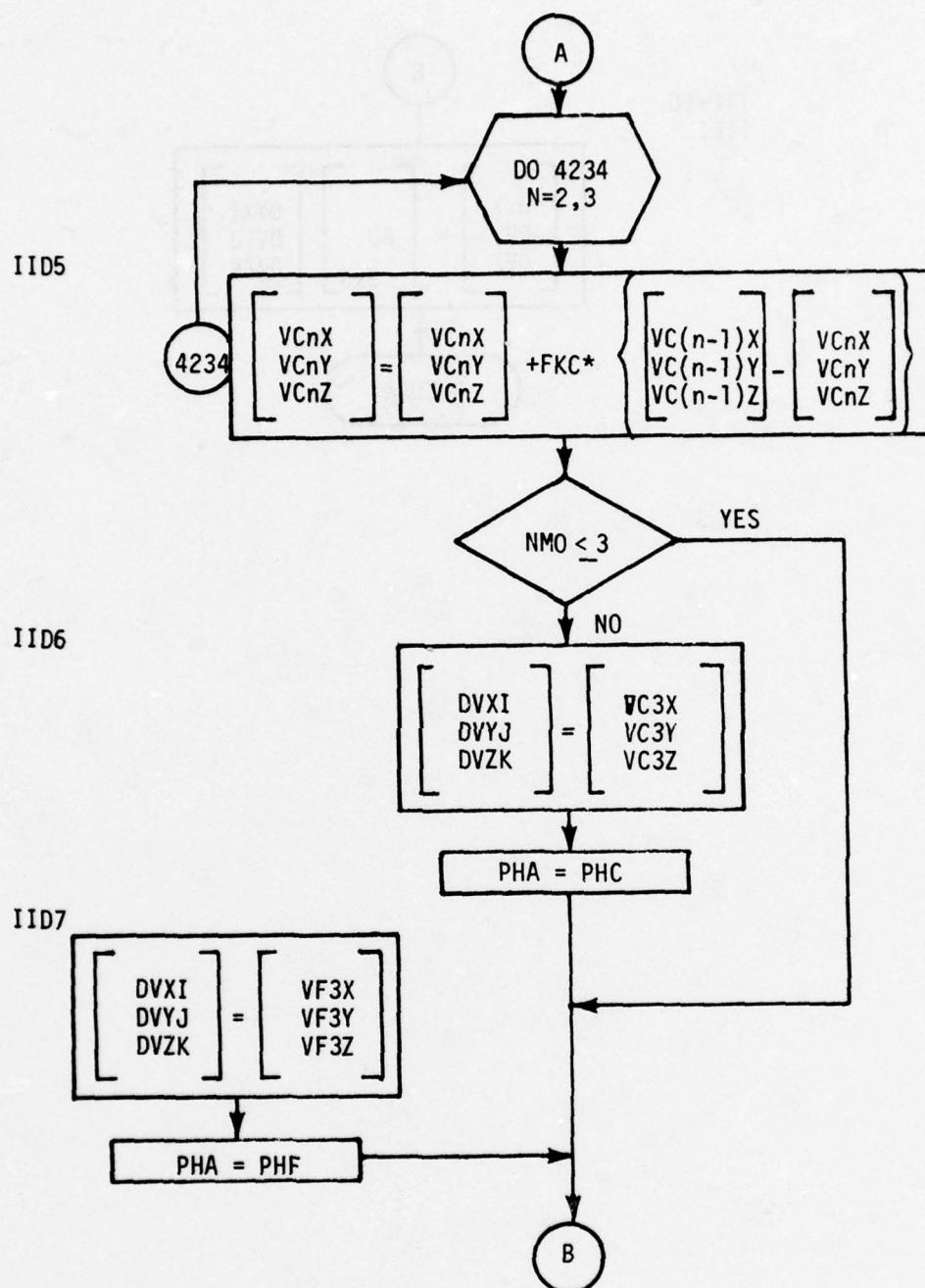
YES  
NMO  $\geq 7$

IID7

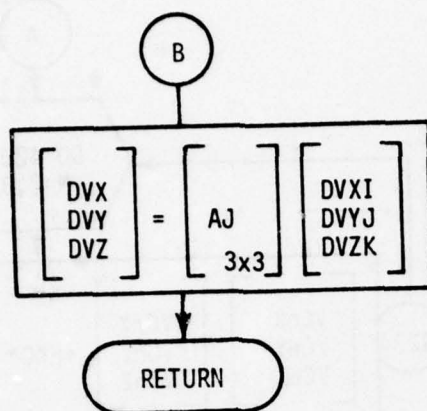
IID4

$$\begin{bmatrix} VCiX \\ VCiY \\ VCiZ \end{bmatrix} = \begin{bmatrix} VCiX \\ VCiY \\ VCiZ \end{bmatrix} + FKC * \left\{ \begin{bmatrix} DVXI \\ DVG \\ DVZK \end{bmatrix} - \begin{bmatrix} VCiX \\ VCiY \\ VCiZ \end{bmatrix} \right\}$$

A



IIE=ID  
IIEI



IIF = IL  
IIF1

IIF2

IIF3

IIF4

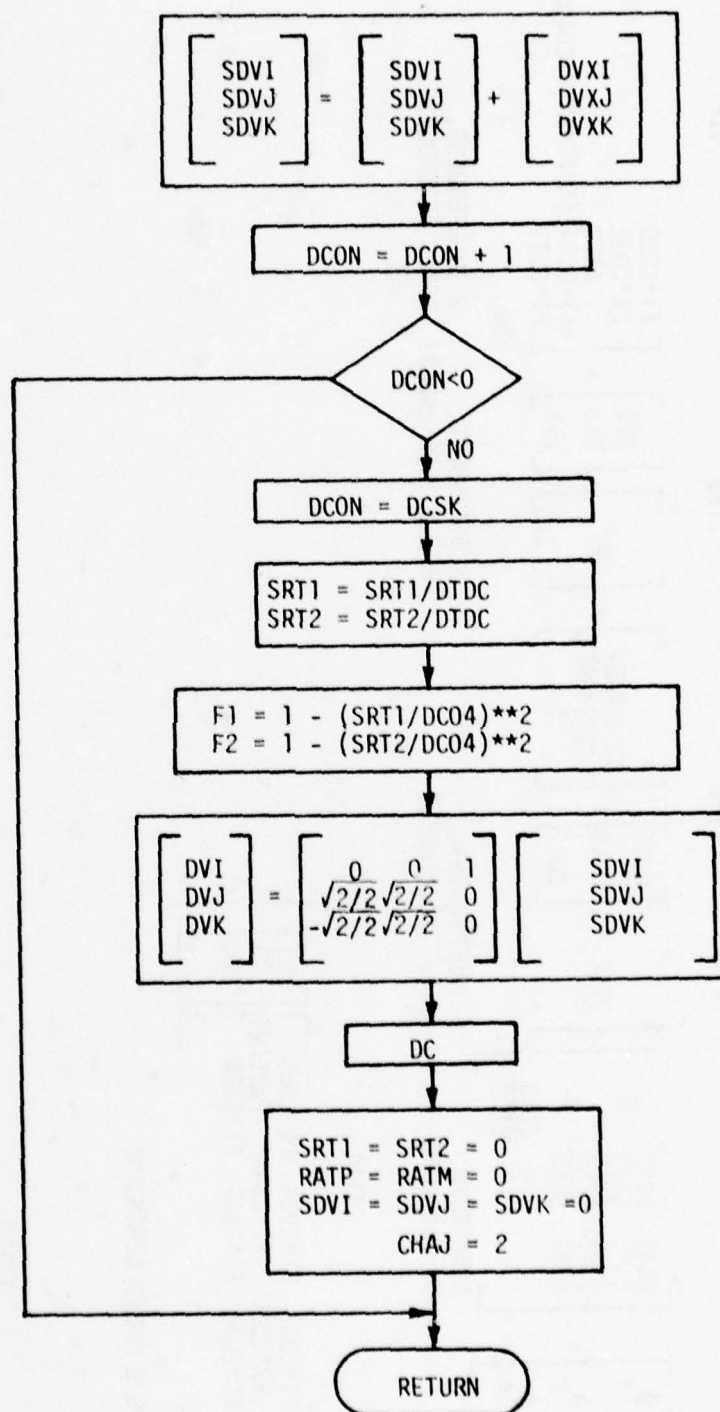
IIF5

IIF6

IIF7

IIF14

IIF15





IIF12

$$\begin{bmatrix} \phi_x \\ \phi_y \\ \phi_z \end{bmatrix} = \begin{bmatrix} \frac{DC42}{SRT1} & \frac{DC42}{SRT2} & 0 \\ \frac{DC42}{SRT1} - \frac{DC42}{SRT2} & \frac{DC42}{SRT2} & 0 \\ 0 & 0 & \frac{DC43}{SRT2} \end{bmatrix}$$

TRANSFORM  
TO PLATFORM

IIF8

$$\left\{ \begin{matrix} DTDC * \\ \end{matrix} \right\}$$

$$\begin{bmatrix} CD16 \\ CD18 \\ CD17 \end{bmatrix} + \begin{bmatrix} 0 \\ CD28*RATP \\ -CD29*RATM \\ 0 \end{bmatrix} +$$

$$\begin{bmatrix} GM \\ 3 \times 3 \end{bmatrix} +$$

$$\begin{bmatrix} DVI \\ DVJ \\ DVK \end{bmatrix} +$$

$$\begin{bmatrix} F1*CD30 \\ F2*CD32 \\ SRT2 \\ +(1-DC04)(CD42*RATP-CD43*RATM) \\ F1*CD31 \end{bmatrix}$$

IIF10

IIF9

G INDEPENDENT,  
SPEED INDEPENDENTG DEPENDENT,  
SPEED INDEPENDENTG INDEPENDENT,  
SPEED DEPENDENT

IIF11

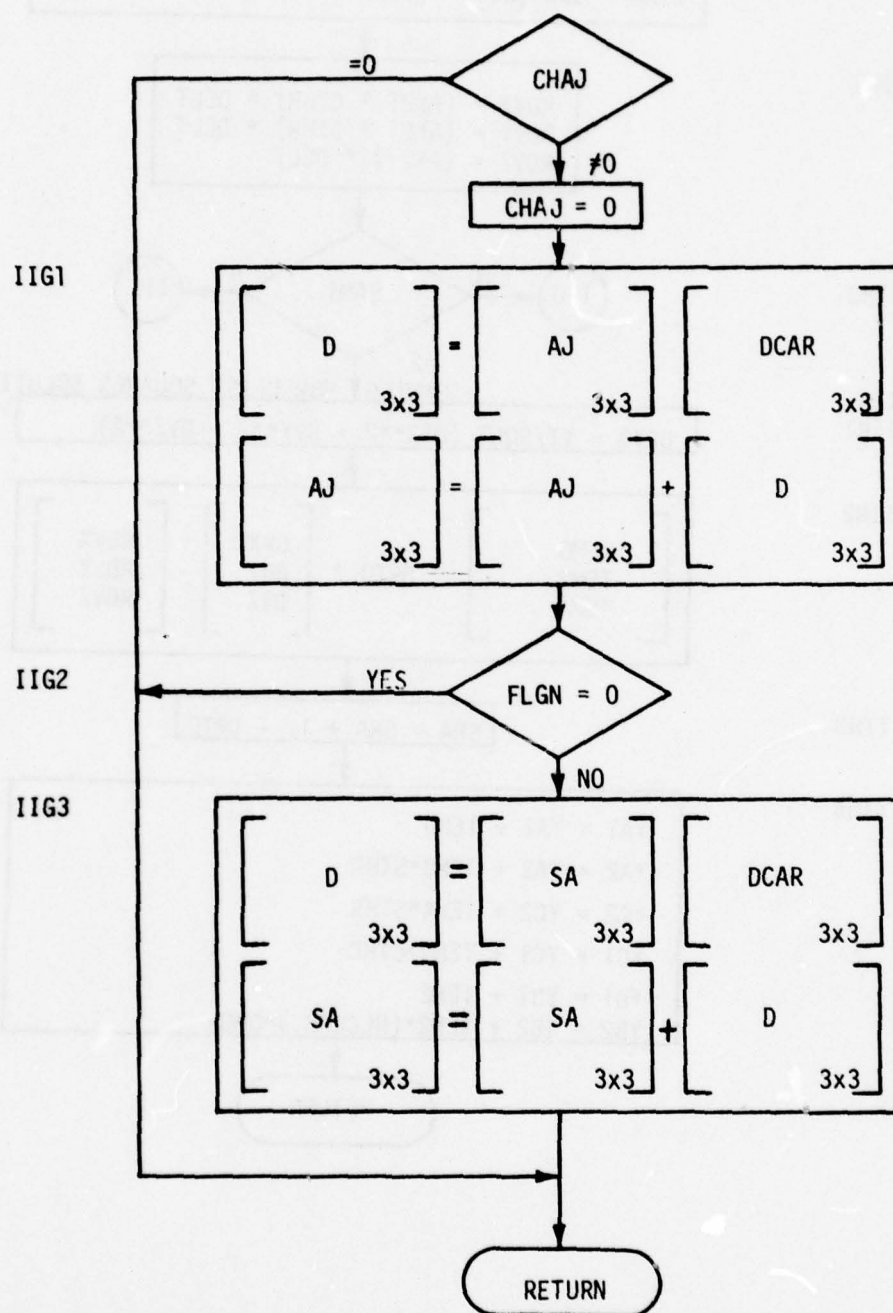
$$\begin{bmatrix} F1*CD33 & F1*CD34 & F1*CD35 \\ -F2*CD39 & -F2*CD40 & -F2*CD41 \\ F1*CD36 & F1*CD37 & F1*CD38 \end{bmatrix} \begin{bmatrix} DVI \\ DVJ \\ DVK \end{bmatrix}$$

G &amp; SPEED DEPENDENT

IIF13

$$\begin{bmatrix} DCAR \\ 3 \times 3 \end{bmatrix} = \begin{bmatrix} 0 & Q_z & \phi_y \\ -\phi_z & 0 & -\phi_x \\ -\phi_y & Q_x & 0 \end{bmatrix}$$

IIG = IM



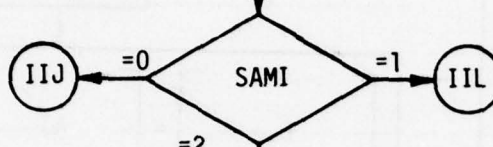
IIH  
IIH1

STHR = SIN (WDT \* (NCCU - .5) - PHA)  
CTHR = COS (WDT \* (NCCU - .5) - PHA)

IIH2

RDVX = (AK2T \* CTHR) \* DELT  
RDVY = (AK2T \* STHR) \* DELT  
RDVZ = (AK2T) \* DELT

IIH3



IIIN  
IIIN1

SUMMING FOR LEAST SQUARES SOLUTION

DPTO = VT/SQRT (DVX\*\*2 + DVY\*\*2 + DVZ\*\*2)

IIIN2

$$\begin{bmatrix} \text{TEM0} \\ \text{TEM2} \\ \text{TEM4} \end{bmatrix} = \text{DPTO} * \begin{bmatrix} \text{DVX} \\ \text{DVY} \\ \text{DVZ} \end{bmatrix} - \begin{bmatrix} \text{RDVX} \\ \text{RDVY} \\ \text{RDVZ} \end{bmatrix}$$

IIIN3

SRA = SRA + 1. - DPTO

IIIN4

YA1 = YA1 + TEM0  
 YA2 = YA2 + TEM0\*STHR  
 YC2 = YC2 + TEM4\*STHR  
 YC1 = YC1 + TEM4\*CTHR  
 YB1 = YB1 + TEM2  
 YB2 = YB2 + TEM2\*(NCCU/8.)\*OMGA

RETURN

SUMMING FOR E.P.A. SOLUTION

IIJ  
IIJ1

$$\begin{bmatrix} VAXI \\ VAYJ \\ VAZK \end{bmatrix} = \begin{bmatrix} VAXI \\ VAYJ \\ VAZK \end{bmatrix} + \begin{bmatrix} DVXI \\ DVYJ \\ DVZK \end{bmatrix}$$

IIJ2

$$\begin{bmatrix} VAX \\ VAY \\ VAZ \end{bmatrix} = \begin{bmatrix} VAX \\ VAY \\ VAZ \end{bmatrix} + \begin{bmatrix} DVX \\ DVY \\ DVZ \end{bmatrix}$$

RETURN

SUMMING FOR LOCAL LEVEL SOLUTION

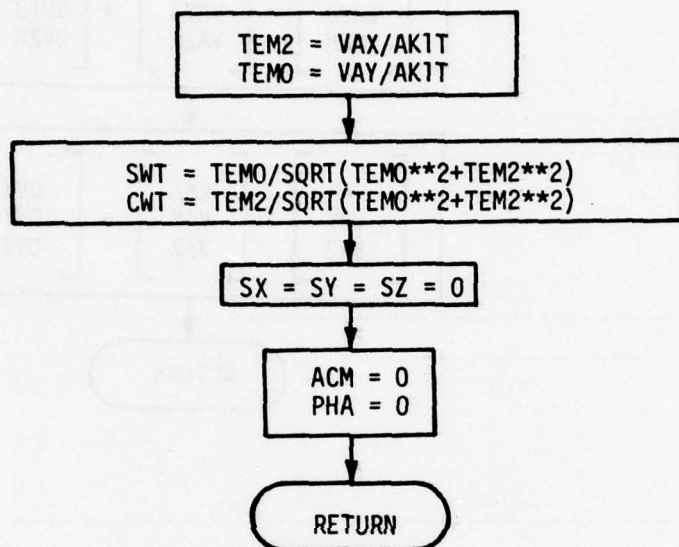
IIIL  
IIIL1

$$\begin{bmatrix} VAX \\ VAY \\ VAZ \end{bmatrix} \begin{bmatrix} VAX \\ VAY \\ VAZ \end{bmatrix} + \begin{bmatrix} DVX \\ DVY \\ DVZ \end{bmatrix} - \begin{bmatrix} RDVX \\ RDVY \\ RDVZ \end{bmatrix}$$

RETURN



## E.P.A. SOLUTION

IIK  
IIK2

## LOCAL LEVEL SOLUTION

IIM  
IIM1

$$\begin{bmatrix} \text{TEM0} \\ \text{TEM2} \\ \text{TEM4} \end{bmatrix} = \frac{1}{\text{VTB}} * \begin{bmatrix} \text{VAX} \\ \text{VAY} \\ \text{VAZ} \end{bmatrix}$$

IIM2

SZ = TEM2\*CGDL  
SX = -TEM2\*SGDL  
SY = -SIGN(TEM4)\*SQRT(TEM4\*\*2+TEM0\*\*2)

ACM = 0

IIM3

SWT = SIN(WOPP\*NCCU)  
CWT = COS(WOPP\*NCCU)

RETURN

## LEAST SQUARES SOLUTION

II0  
II01

$$\begin{bmatrix} XA \\ 2 \times 1 \end{bmatrix} = \begin{bmatrix} a_{MCSI} \\ * \end{bmatrix} \begin{bmatrix} YA \\ 2 \times 1 \end{bmatrix}$$

$$\begin{bmatrix} XB \\ 2 \times 1 \end{bmatrix} = \begin{bmatrix} /b_{MCSI} \\ * \end{bmatrix} \begin{bmatrix} YB \\ 2 \times 1 \end{bmatrix}$$

$$\begin{bmatrix} XC \\ 2 \times 1 \end{bmatrix} = \begin{bmatrix} C_{MCSI} \\ * \end{bmatrix} \begin{bmatrix} YC \\ 2 \times 1 \end{bmatrix}$$

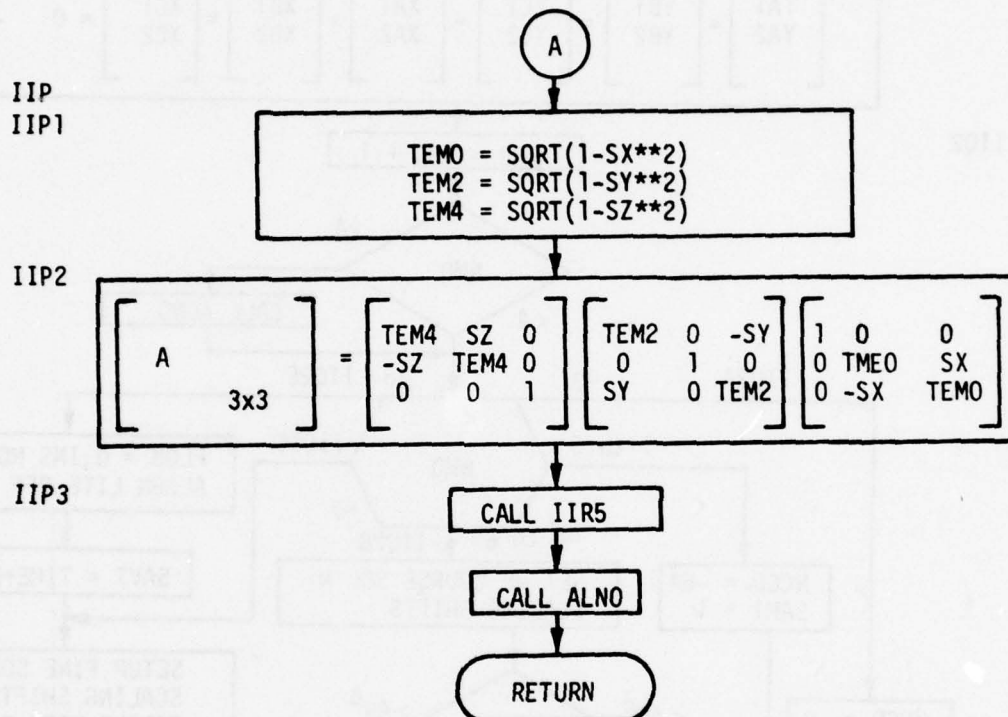
II02

$$\begin{aligned} SX &= XC(2)/AK1T \\ SZ &= XA(2)/AK1T \\ SY &= -XC(1)/AK1T \\ ACM &= SRA/NCCU \\ SX &= SX + SGDL * (SZ * CGDL - XB(1)/VTC - SX * SGDL) \\ SZ &= SZ - CGDL * (SZ * CGDL - XB(1)/VTC - SX * SGDL) \end{aligned}$$

II04 = IIM3

$$\begin{aligned} SWT &= \sin(WOPP * NCCU) \\ CWT &= \cos(WOPP * NCCU) \end{aligned}$$

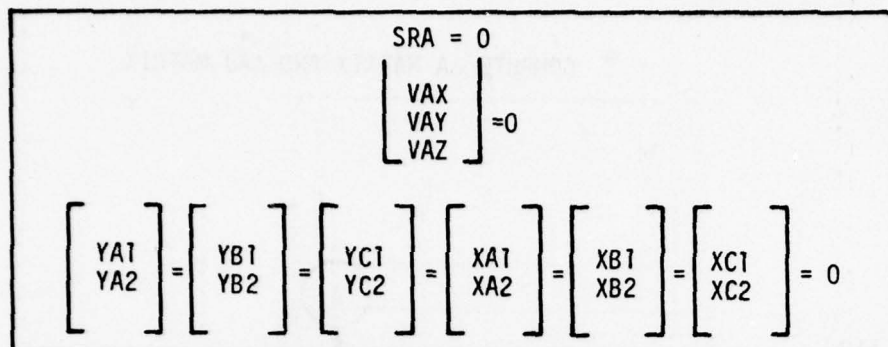
A

COMPUTE  $\Delta A$  MATRIX AND  $\Delta A_J$  MATRIX

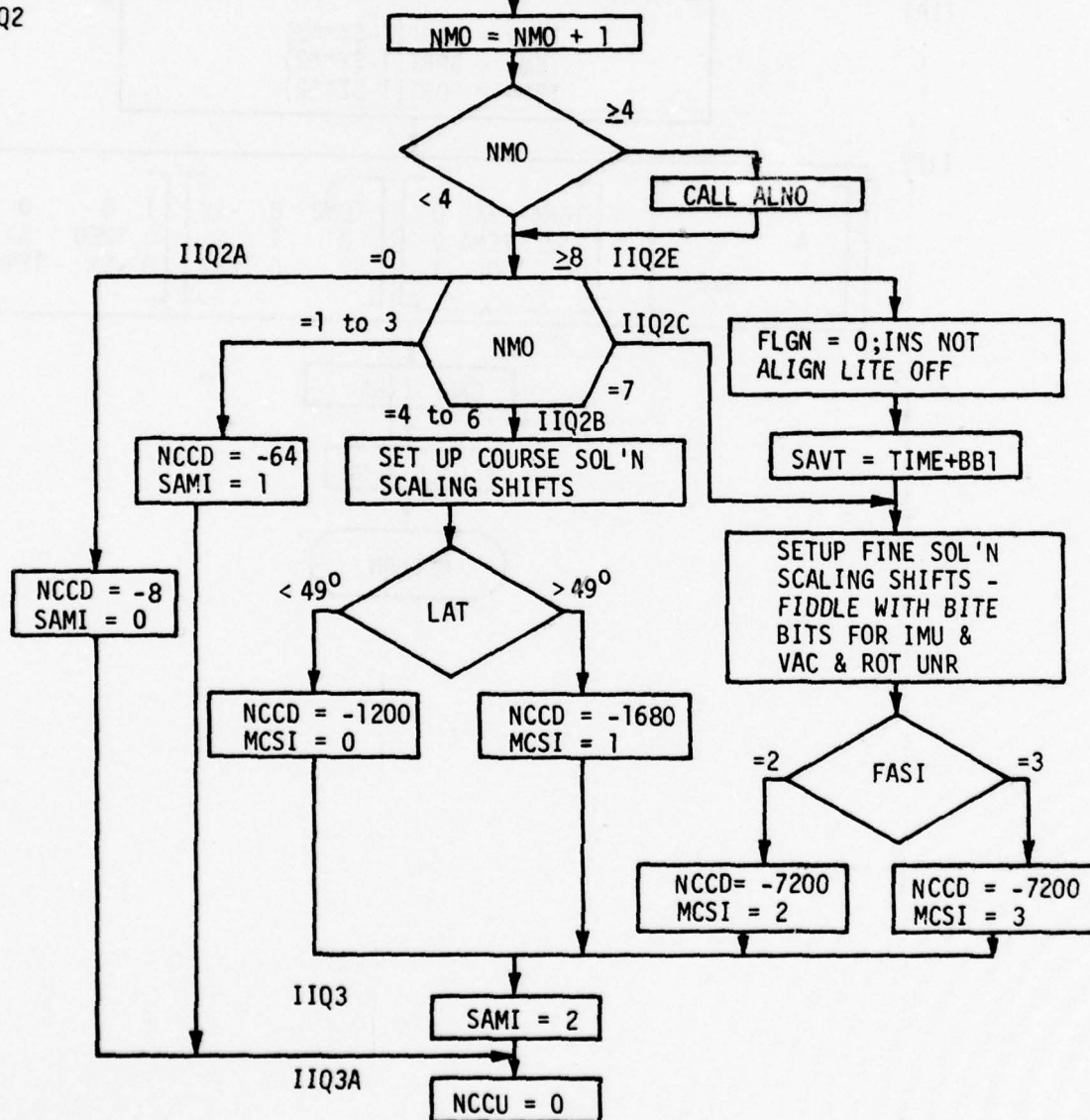


## SUBROUTINE RESET

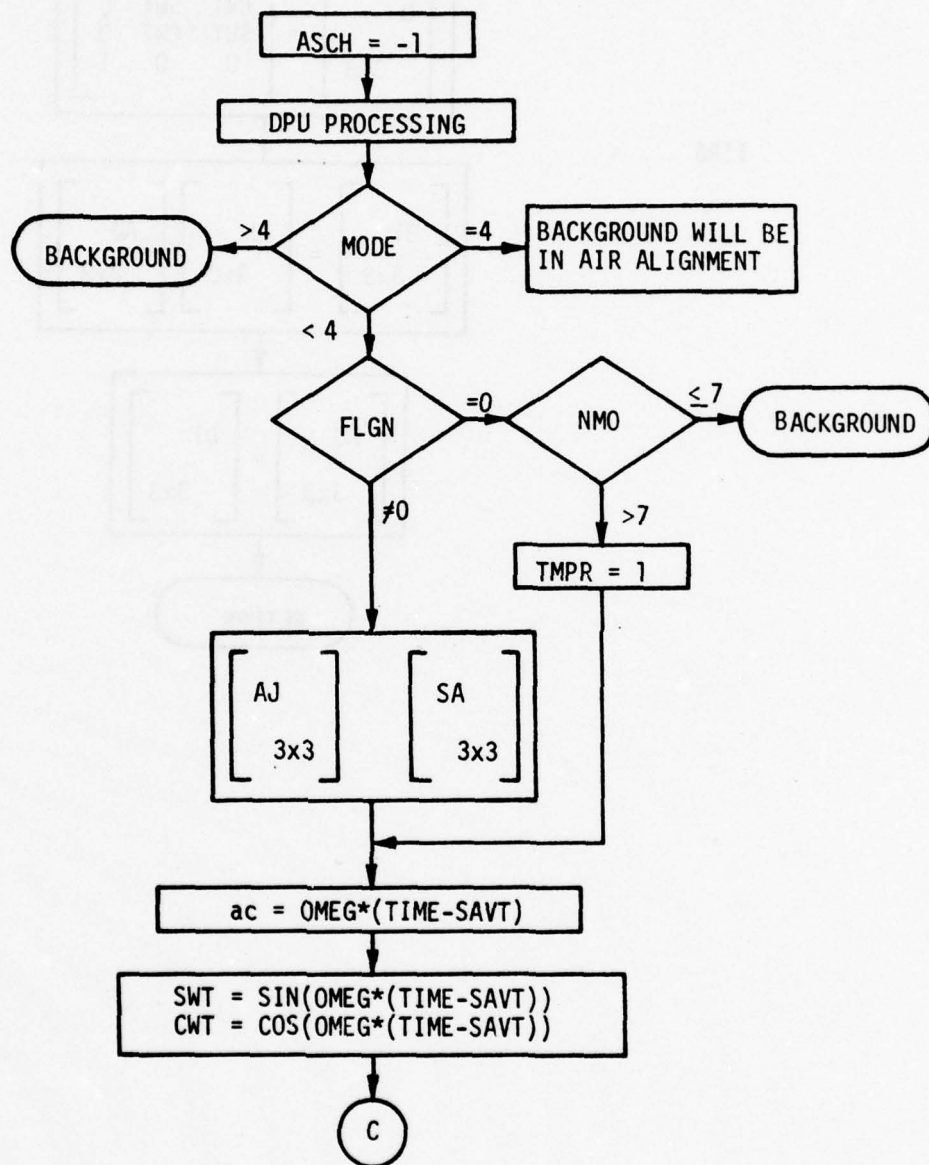
IIQ1



IIQ2



GO TO NAV DECISION

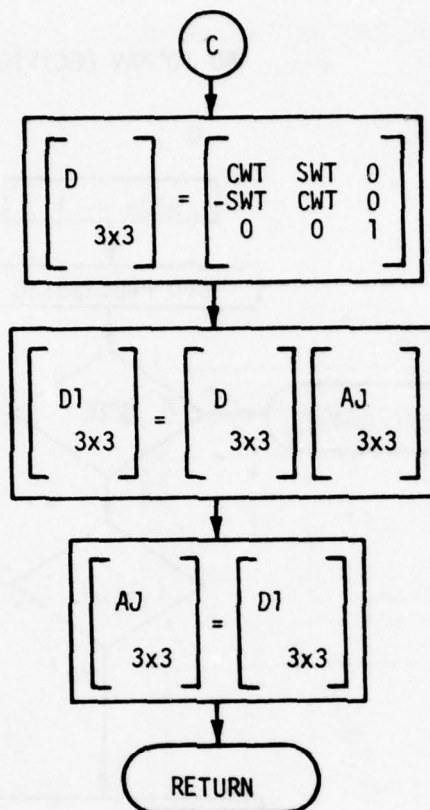
IIR  
IIR1

IIR2

IIR3

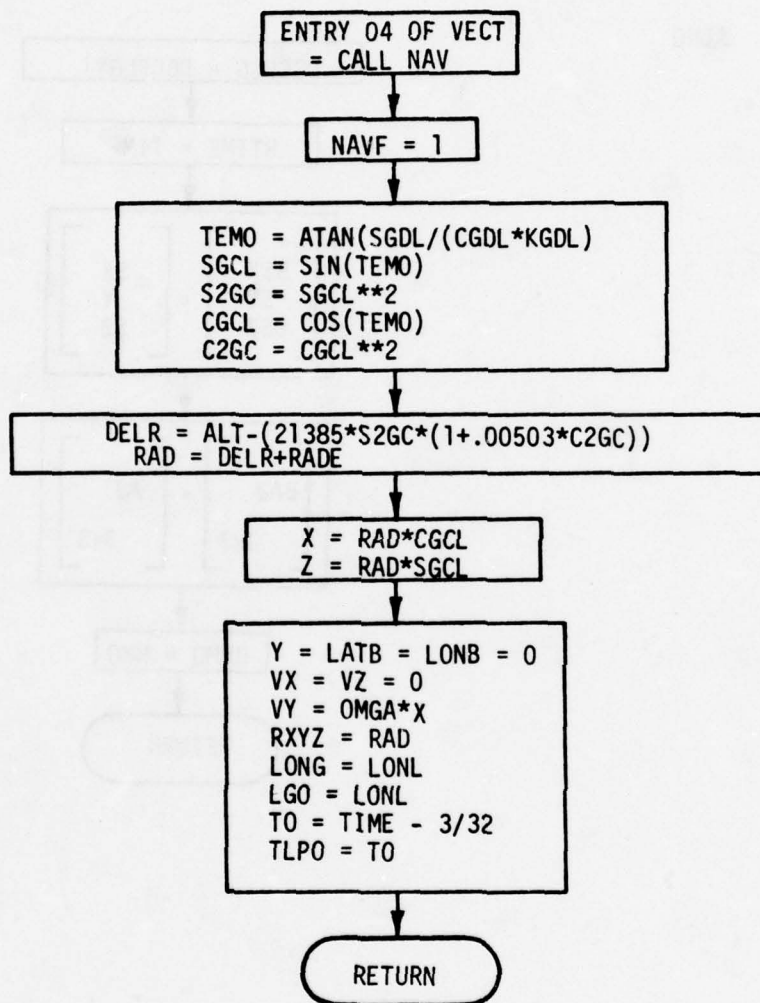
IIR5

IIR6



## NAVIGATION INITIALIZATION

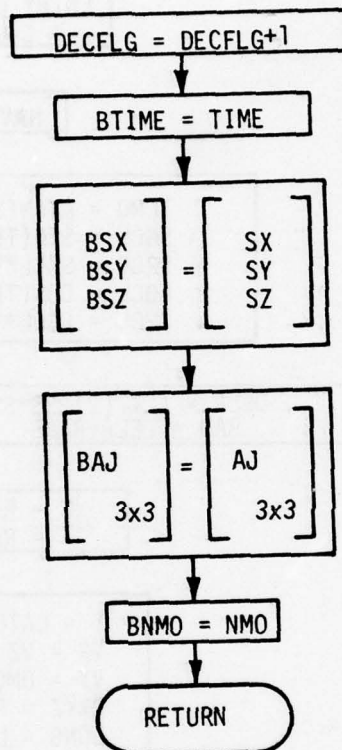
NAVI



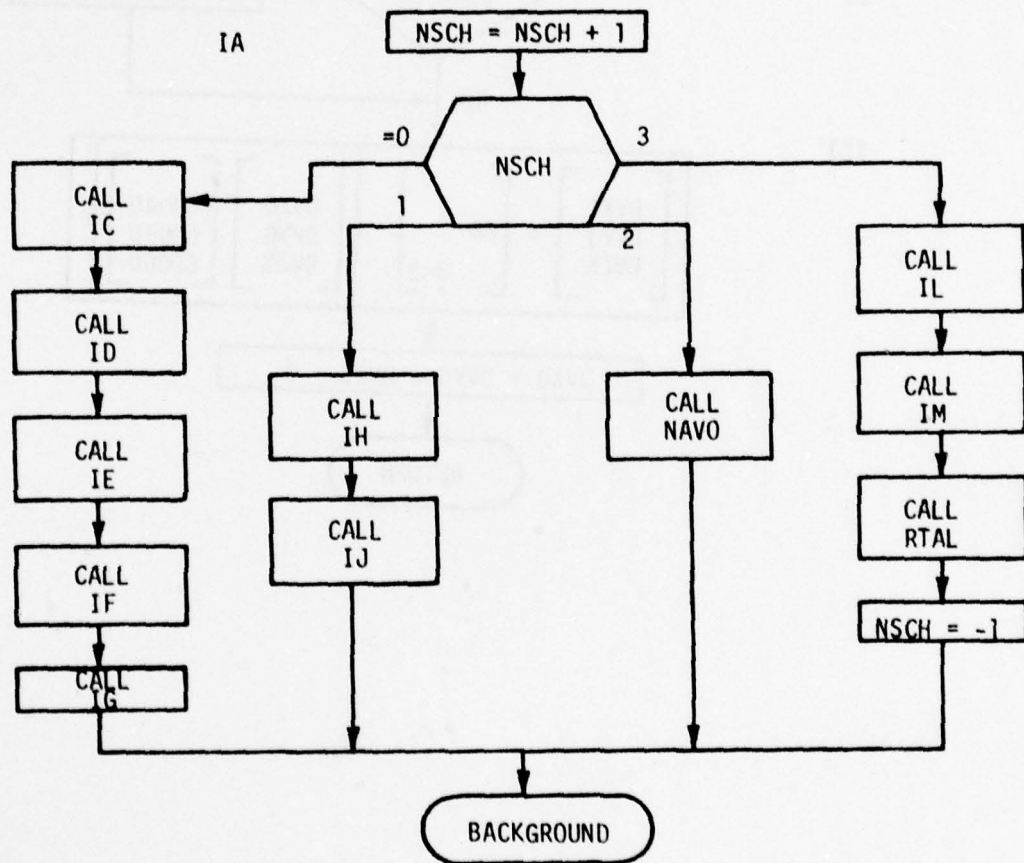


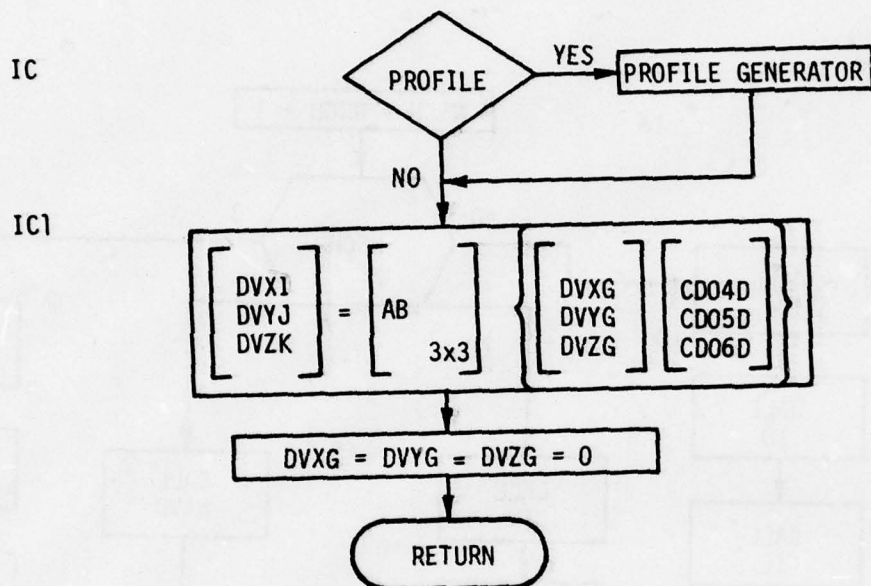
## ALIGNMENT OUTPUT ROUTINE

ALNO



## NAVIGATION SUB-EXECUTIVE



ACCELEROMETER BIAS & SCALE FACTOR COMPUTATION  
AND NON-ORTHOGONALITY COMPENSATION

AFAL-TR-77-119

ROTATION FROM PLATFORM FRAME TO NAVIGATION FRAME

ID  
ID1

$$\begin{bmatrix} DVX \\ DVY \\ DVZ \end{bmatrix} = \begin{bmatrix} AJ \\ 3 \times 3 \end{bmatrix} \begin{bmatrix} DVXI \\ DVYJ \\ DVZK \end{bmatrix}$$

RETURN

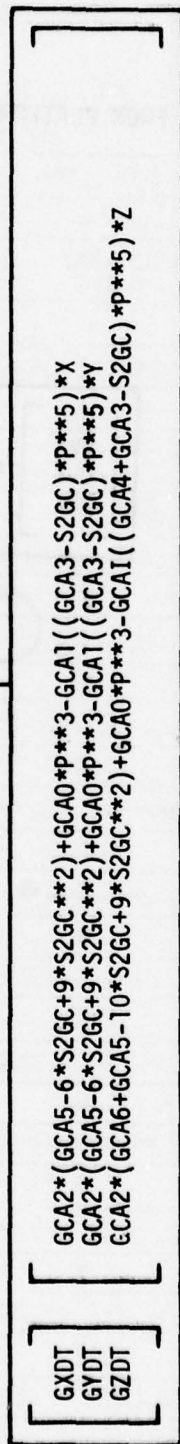


GRAVITY MODEL

IE  
IE1

IE2

$$P = 1 - (\text{DEL R}/\text{RADE}) + (\text{DEL R}/\text{RADE})^{**2}$$



## VERTICAL DAMPING COMPUTATION

IF  
IF2

$$\text{DEL R} = \text{ALT} - (21385 * 52\text{GC} * (1 + .00503 * \text{C2GC}))$$

IF3

$$\text{RAD} = \text{DEL R} + \text{RADE}$$

IF1

$$\text{Ø46 DATA} = \text{ALT} - (\text{RAD} - \text{RXYZ})$$

$$\begin{bmatrix} \text{LDVX} \\ \text{LDVY} \\ \text{LDVZ} \end{bmatrix} = \text{CD52} * (\text{RAD} - \text{RXYZ}) * \begin{bmatrix} \text{X} \\ \text{Y} \\ \text{Z} \end{bmatrix}$$

RETURN

## DOUBLE INTEGRATION FOR VELOCITY AND DISTANCE

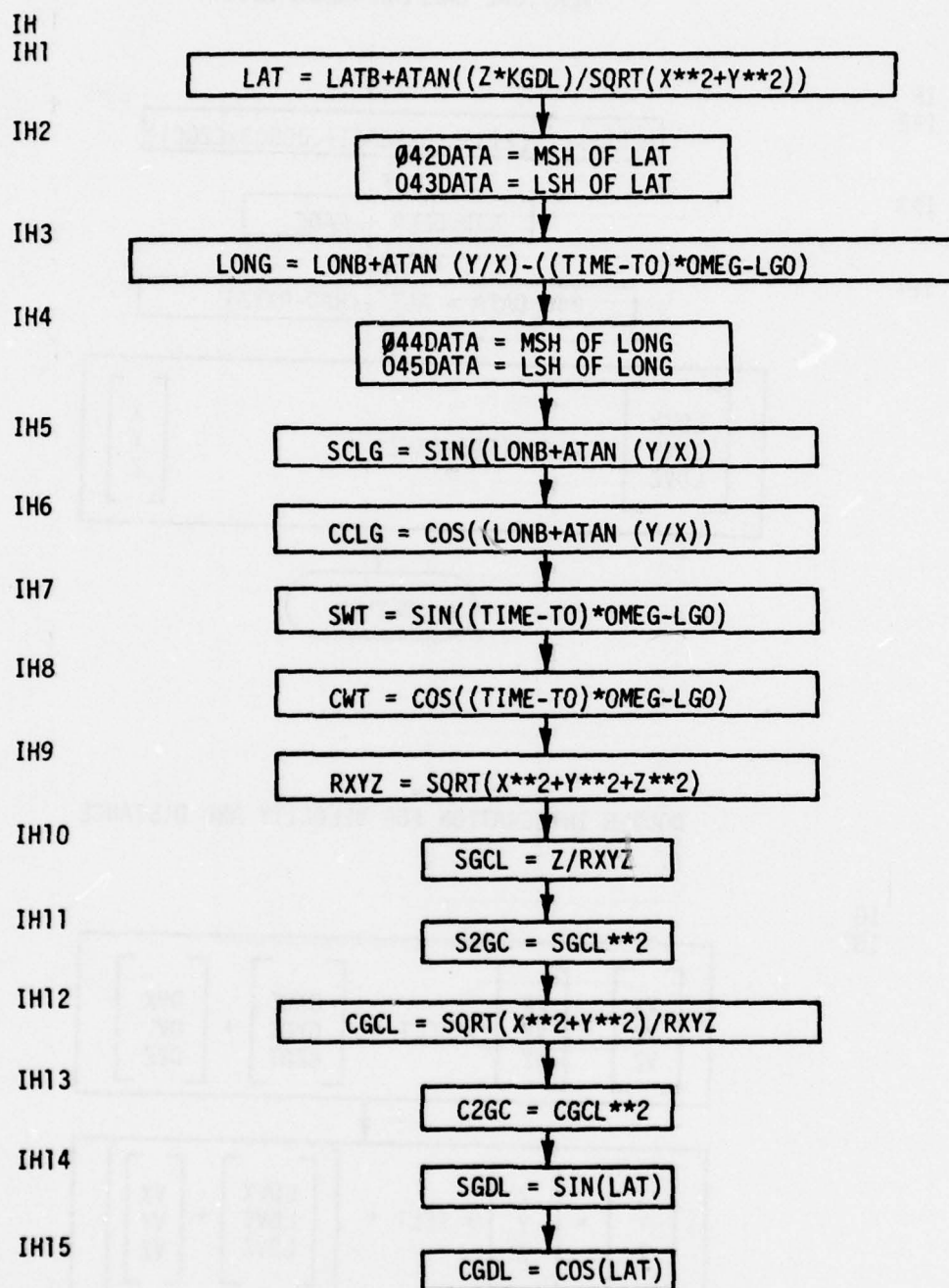
IG  
IG1

$$\begin{bmatrix} \text{VX} \\ \text{VY} \\ \text{VZ} \end{bmatrix} = \begin{bmatrix} \text{VX} \\ \text{VY} \\ \text{VZ} \end{bmatrix} + \text{DELT} * \begin{bmatrix} \text{GXDT} \\ \text{GYDT} \\ \text{GZDT} \end{bmatrix} + \begin{bmatrix} \text{DVX} \\ \text{DVY} \\ \text{DVZ} \end{bmatrix}$$

$$\begin{bmatrix} \text{X} \\ \text{Y} \\ \text{Z} \end{bmatrix} = \begin{bmatrix} \text{X} \\ \text{Y} \\ \text{Z} \end{bmatrix} + \text{DELT} * \left\{ \begin{bmatrix} \text{LDVX} \\ \text{LDVY} \\ \text{LDVZ} \end{bmatrix} + \begin{bmatrix} \text{VX} \\ \text{VY} \\ \text{VZ} \end{bmatrix} \right\}$$

RETURN

## LATITUDE AND LONGITUDE COMPUTATION



## LOCAL VERTICAL CO-ORDINATES AND GROUND SPEED

IJ

IJ1

$$\begin{bmatrix} VV \\ VE \\ VN \end{bmatrix} = \begin{bmatrix} CGDL * CCLG & CGDL * SCLG & SGDL \\ -SCLG & CCLG & 0 \\ -SGDL * CCLG & -SGDL * SCLG & CGDL \end{bmatrix} \begin{bmatrix} VX + Y * OMGA \\ VY - X * OMGA \\ VZ \end{bmatrix}$$

IJ2

$$\begin{bmatrix} VXE \\ VYE \\ VZE \end{bmatrix} = \begin{bmatrix} CWT & SWT & 0 \\ -SWT & CWT & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} VX + Y * OMGA \\ VY - X * OMGA \\ VZ \end{bmatrix}$$

IJ3

$$\begin{bmatrix} \text{Ø48} \\ \text{Ø4A} \\ \text{Ø4C} \end{bmatrix} = \begin{bmatrix} \text{LSH OF } VV \\ \text{LSH OF } VE \\ \text{LSH OF } VN \end{bmatrix}$$

$$\begin{bmatrix} \text{Ø47} \\ \text{Ø49} \\ \text{Ø4B} \end{bmatrix} = \begin{bmatrix} \text{MSH OF } VV \\ \text{MSH OF } VE \\ \text{MSH OF } VN \end{bmatrix}$$

IJ4

$$VEL2 = \begin{bmatrix} VV \\ VE \\ VN \end{bmatrix} \cdot \begin{bmatrix} VV \\ VE \\ VN \end{bmatrix}$$

IJ5

$$GS = \text{SQRT} (VEL2 - VV * 2)$$

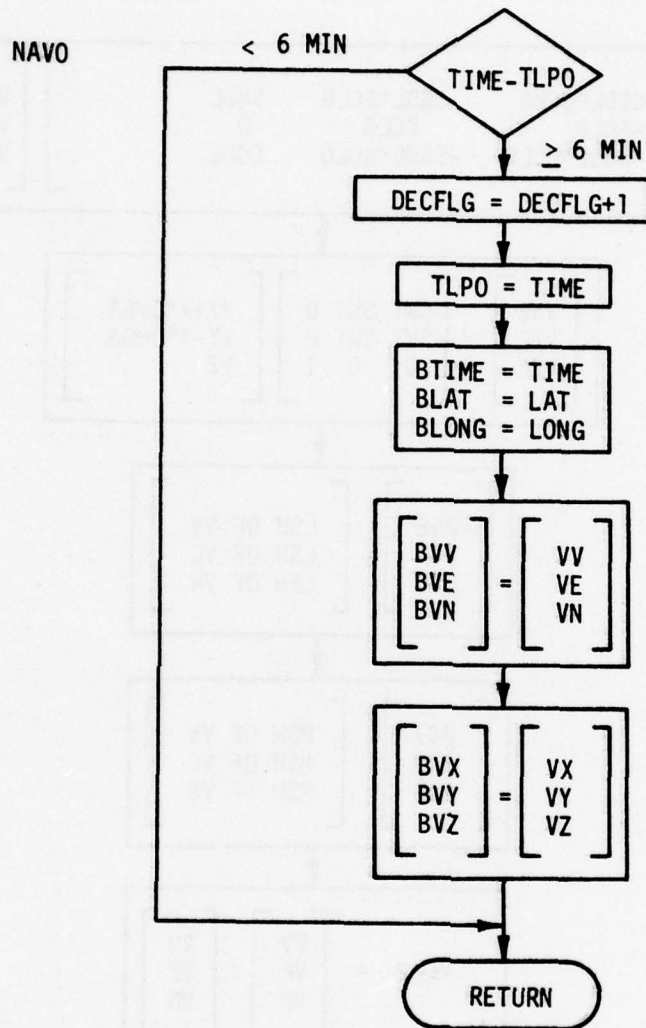
IJ6

$$\begin{bmatrix} V1 \\ 3 \times 3 \end{bmatrix} = \begin{bmatrix} CEDL * CCLG & CGDL * SCLG & SGDL \\ -SCLG & CCLG & 0 \\ -SGDL * CCLG & -SGDL * SCLG & CGDL \end{bmatrix} \begin{bmatrix} AJ \\ 3 \times 3 \end{bmatrix}$$

RETURN



# NAVIGATION OUTPUT ROUTINE



## DRIFT COMPENSATION

IL  
IL1

$$\begin{bmatrix} \text{SDVI} \\ \text{SDVJ} \\ \text{SDVK} \end{bmatrix} = \begin{bmatrix} \text{SDVI} \\ \text{SDVJ} \\ \text{SDVK} \end{bmatrix} + \begin{bmatrix} \text{DVXI} \\ \text{DVXJ} \\ \text{DVXK} \end{bmatrix}$$

IL2

DCON = DCON+1

IL3

YES  
DCON < 0

IL4

NO

DCON = DCSK

IL5

$$\begin{aligned} \text{SRT1} &= \text{SRT1}/\text{DTDC} \\ \text{SRT2} &= \text{SRT2}/\text{DTDC} \end{aligned}$$

IL6

$$\begin{aligned} \text{F1} &= 1 - (\text{SRT1}/\text{DC04})^{**2} \\ \text{F2} &= 1 - (\text{SRT2}/\text{DC04})^{**2} \end{aligned}$$

IL7

$$\begin{bmatrix} \text{DVI} \\ \text{DVJ} \\ \text{DJK} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ \sqrt{2}/2 & \sqrt{2}/2 & 0 \\ -\sqrt{2}/2 & \sqrt{2}/2 & 0 \end{bmatrix} \begin{bmatrix} \text{SDVI} \\ \text{SDVJ} \\ \text{SDVK} \end{bmatrix}$$

IL8 --  
IL13D C  
SEE NEXT PAGE

IL14

$$\begin{aligned} \text{SRT1} &= \text{SRT2} = 0 \\ \text{RATP} &= \text{RATM} = 0 \\ \text{SDVI} &= \text{SDVJ} = \text{SDVK} = 0 \end{aligned}$$

IL15

CHAJ = 0

RETURN

$$\begin{aligned}
 & \begin{bmatrix} \phi_x \\ \phi_y \\ \phi_z \end{bmatrix} = \begin{bmatrix} \frac{DC42}{SRT1} & \frac{DC42}{SRT2} & 0 \\ \frac{DC42}{SRT1} & \frac{DC42}{SRT2} & 0 \\ 0 & 0 & \frac{DC43}{SRT2} \end{bmatrix} \left\{ \begin{array}{c} DTDC * \\ \\ \end{array} \right\} \begin{bmatrix} CD16 \\ CD18 \\ CD17 \end{bmatrix} + \begin{bmatrix} 0 \\ CD28 * RATP \\ -CD29 * RATM \\ 0 \end{bmatrix} + \begin{bmatrix} GM \\ \\ \\ \end{bmatrix}_{3 \times 3} + \begin{bmatrix} DVI \\ DVJ \\ DVK \end{bmatrix} + \begin{bmatrix} F1 * CD30 \\ F2 * CD32 \\ SRT2 \\ +(1-DC04)(CD42 * RATP - CD43 * RATM) \\ F1 * CD31 \end{bmatrix} + \\
 & \text{IL12} \qquad \text{IL11} \qquad \text{IL10}
 \end{aligned}$$

TRANSFORM  
TO PLATFORM

G INDEPENDENT,  
SPEED INDEPENDENT

G DEPENDENT,  
SPEED INDEPENDENT

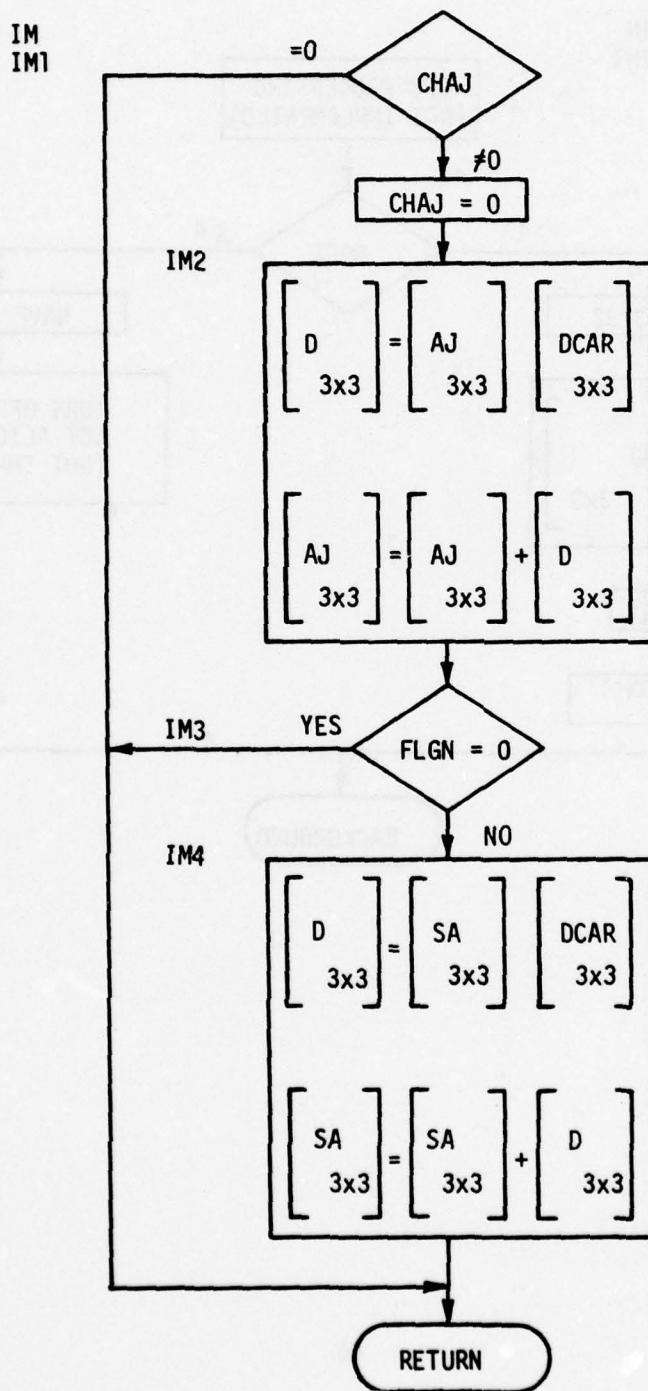
G INDEPENDENT,  
SPEED DEPENDENT

$$\begin{aligned}
 & \begin{bmatrix} F1 * CD33 & F1 * CD34 & F1 * CD35 \\ -F2 * CD39 & -F2 * CD40 & -F2 * CD41 \\ F1 * CD36 & F1 * CD37 & F1 * CD38 \end{bmatrix} \begin{bmatrix} DVI \\ DVJ \\ DVK \end{bmatrix} \\
 & \text{IL11}
 \end{aligned}$$

G & SPEED DEPENDENT

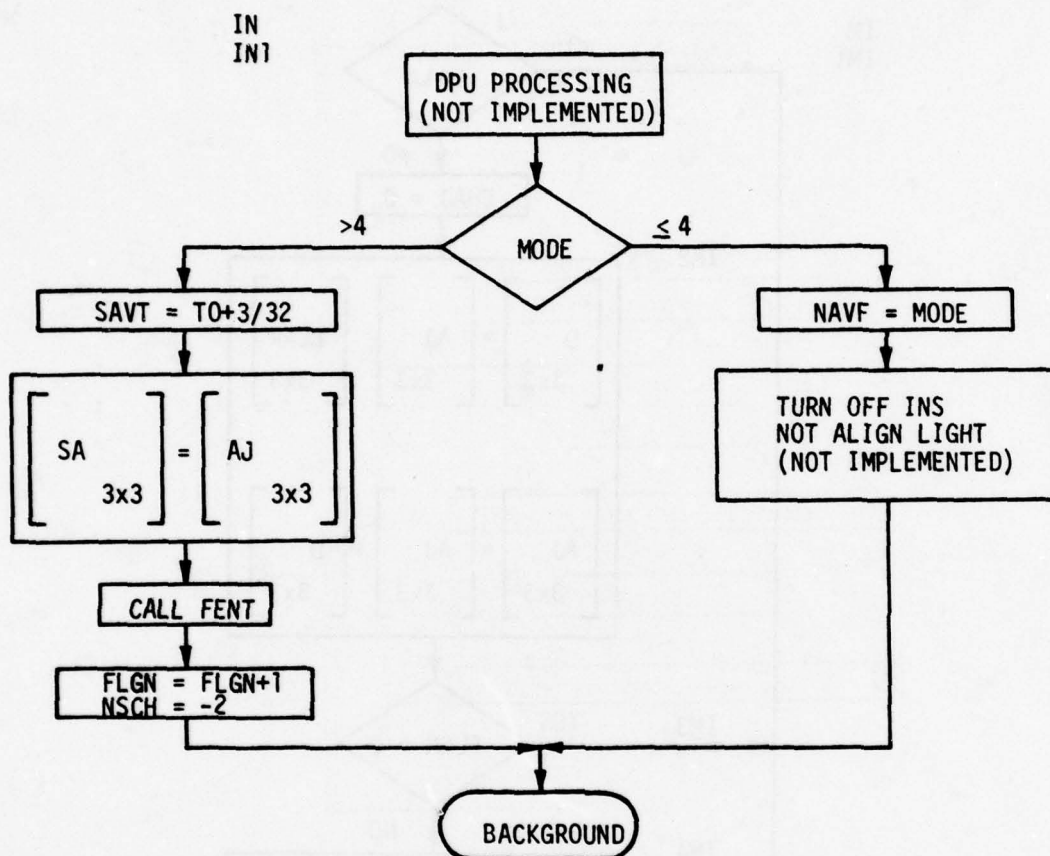
$$\begin{aligned}
 & \begin{bmatrix} DCAR \\ \\ \end{bmatrix}_{3 \times 3} = \begin{bmatrix} 0 & Q_z & \phi_y \\ -\phi_z & 0 & -\phi_x \\ -\phi_y & Q_x & 0 \end{bmatrix} \\
 & \text{IL13}
 \end{aligned}$$

## UPDATE AJ AND SA MATRICES



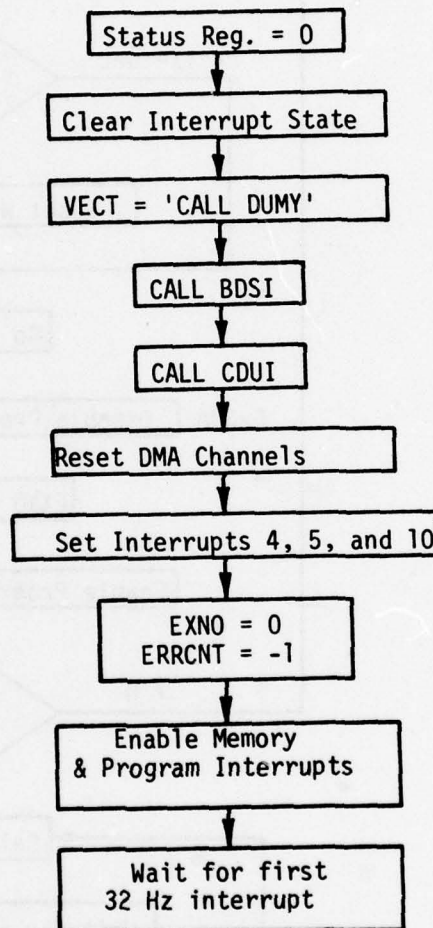


## RETURN TO ALIGN DECISION (RTAL)



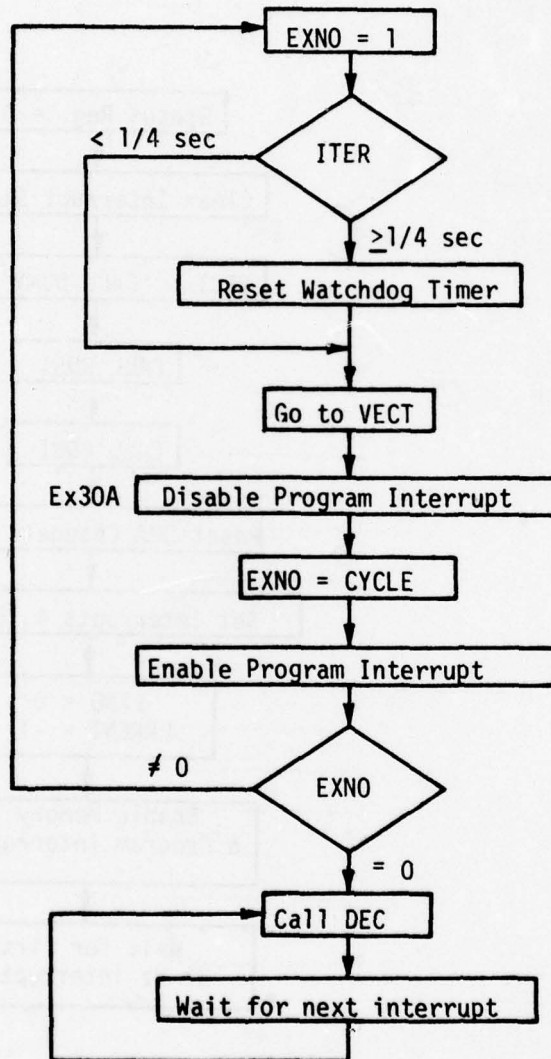
## EXECUTIVE INITIALIZATION

EXEC



## SKC-2000 Executive

Ex30



Vector Table

During Alignment

VECT

```
CALL DUMY
CALL DECD
CALL CDU
CALL ALIGN (IIA)
CALL SPIN (DUMY)
CALL DUMY
CALL BITE (DUMY)
CALL DUMY
CALL DUMY
CALL DUMY
CALL DUMY
CALL DUMY
CALL DUMY
CALL GASC (DUMY)
GO TO Ex30A
```

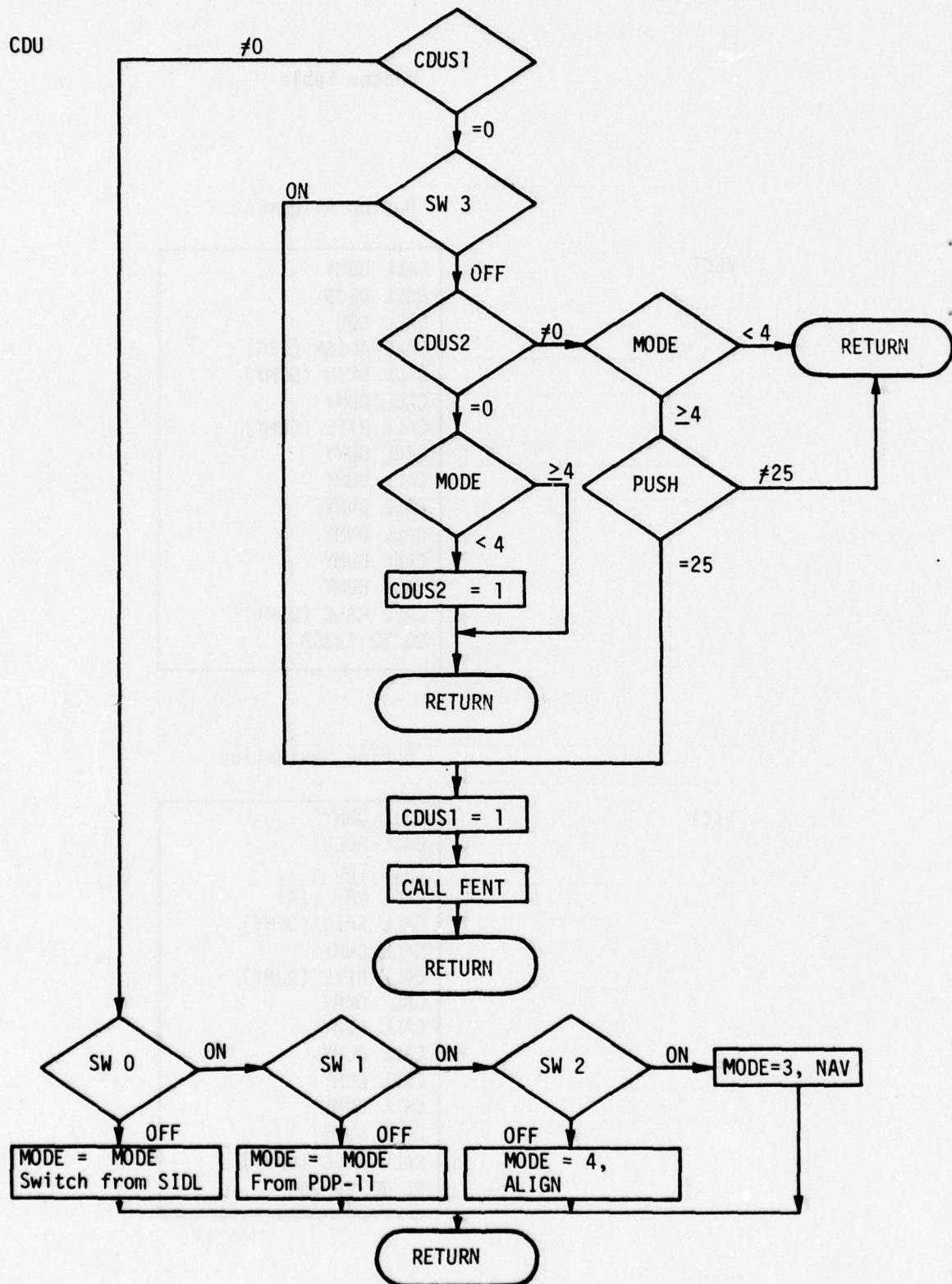
During Navigation

VECT

```
CALL DUMY
CALL DECD
CALL CDU
CALL NAV (IA)
CALL SPIN (DUMY)
CALL DUMY
CALL BITE (DUMY)
CALL DUMY
CALL DUMY
CALL DUMY
CALL DUMY
CALL DUMY
CALL DUMY
CALL GASC (DUMY)
GO TO Ex 30A
```



Synchronize SKC-2000 Alignment with Honeywell Alignment.



# CDU Initialization

CDUI

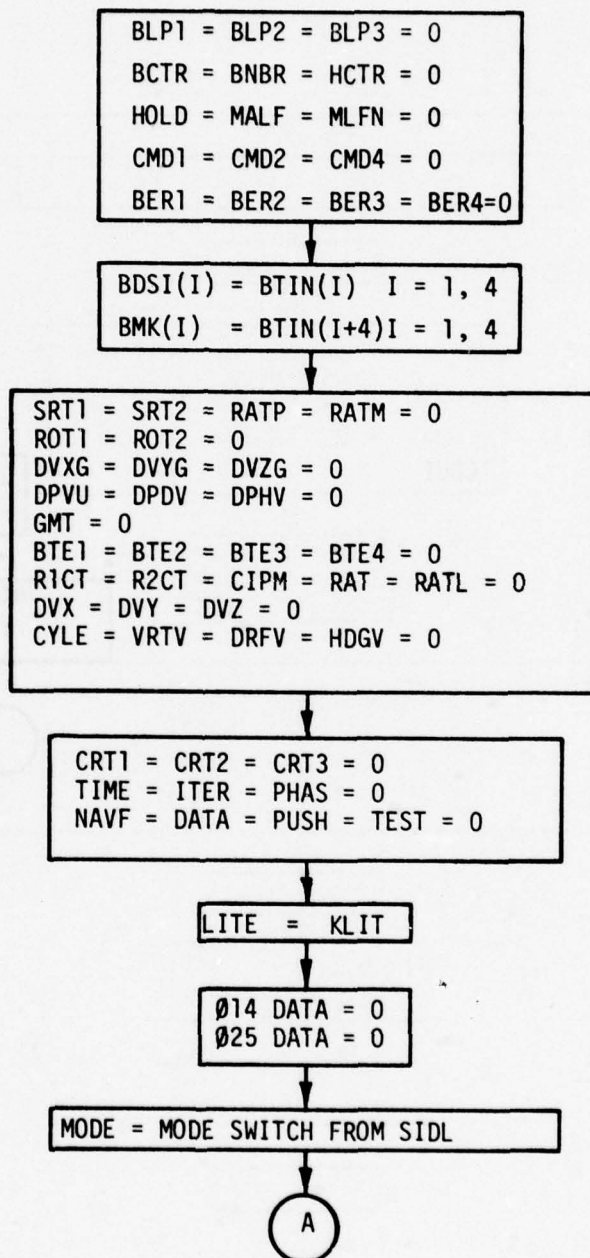
CDUS1 = 0  
CDUS2 = 0

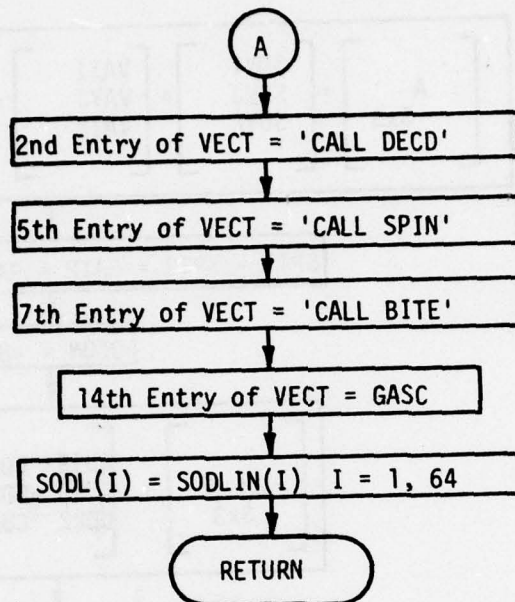
Set 3RD Entry  
of VECT to  
'CALL CDU'

RETURN

Initialize Built-In Test, Data Decode, & Auto Sequencing.

BDSI







## Initialize for Alignment (First Entry)

FENT

$$\begin{bmatrix} A \\ 3 \times 3 \end{bmatrix} = \begin{bmatrix} SDVI \\ SDVJ \\ SDVK \end{bmatrix} = \begin{bmatrix} VAXI \\ VAYJ \\ VAZK \end{bmatrix} = \begin{bmatrix} DVXI \\ DVYJ \\ DVZK \end{bmatrix} = \begin{bmatrix} DVXG \\ DVYG \\ DVZG \end{bmatrix} = 0$$

$$SRT1 = SRT2 = RATP = RATM = CHA1 = 0$$

$$DCON = -8$$

$$\begin{bmatrix} GM \\ 3 \times 3 \end{bmatrix} = \begin{bmatrix} CD19 & CD20 & CD21 \\ CD25 & -CD26 & -CD27 \\ CD22 & CD23 & CD24 \end{bmatrix}$$

$$\begin{aligned} A(1, 1) &= CD01 \\ A(2, 2) &= CD02 \\ A(3, 3) &= CD03 \end{aligned}$$

$$\begin{bmatrix} AB \\ 3 \times 3 \end{bmatrix} = \begin{bmatrix} CD07 & CD10 & CD13 \\ CD08 & CD11 & CD14 \\ CD09 & CD12 & CD15 \end{bmatrix} \begin{bmatrix} A \\ 3 \times 3 \end{bmatrix}$$

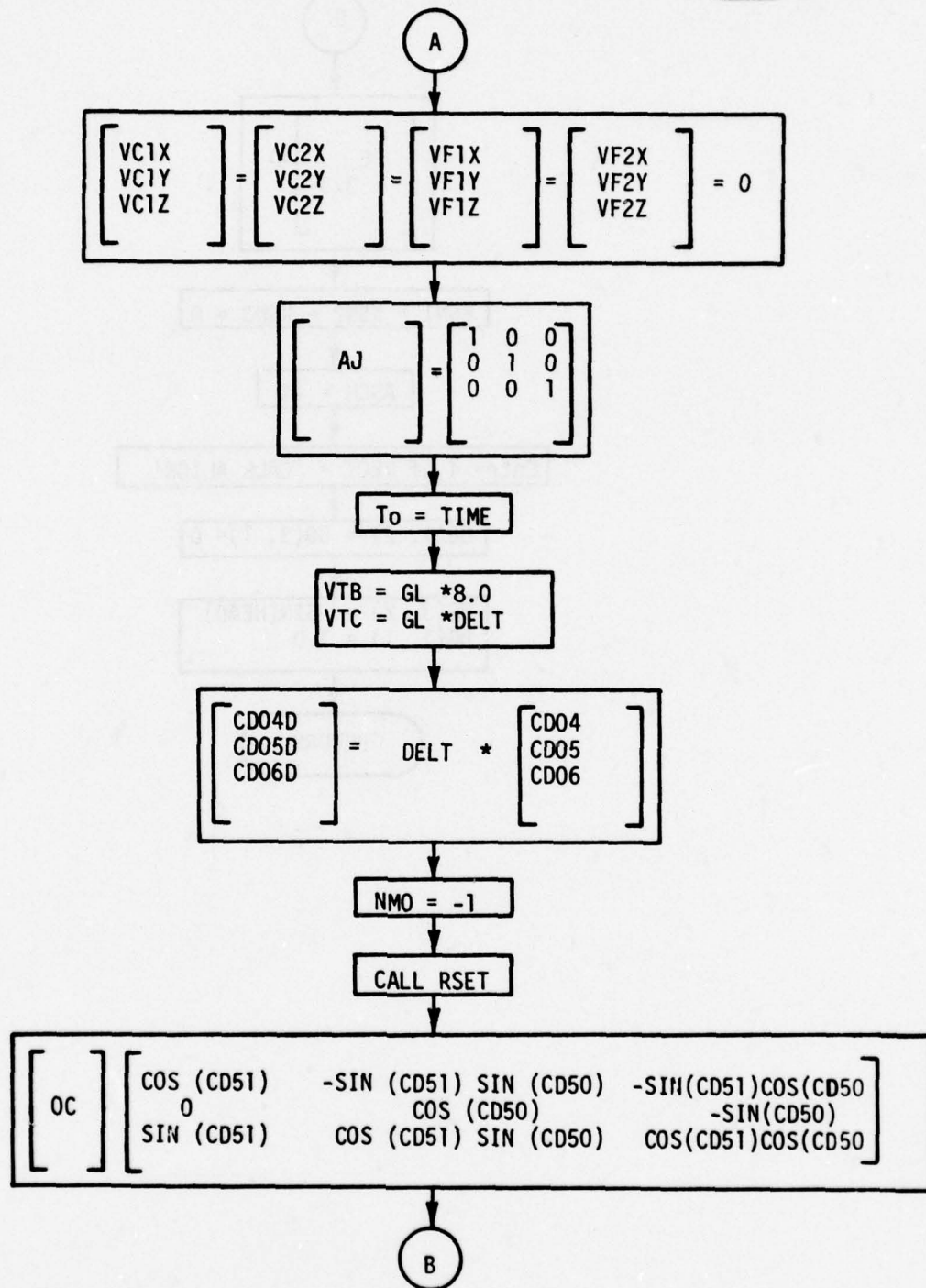
$$LAT = LATL$$

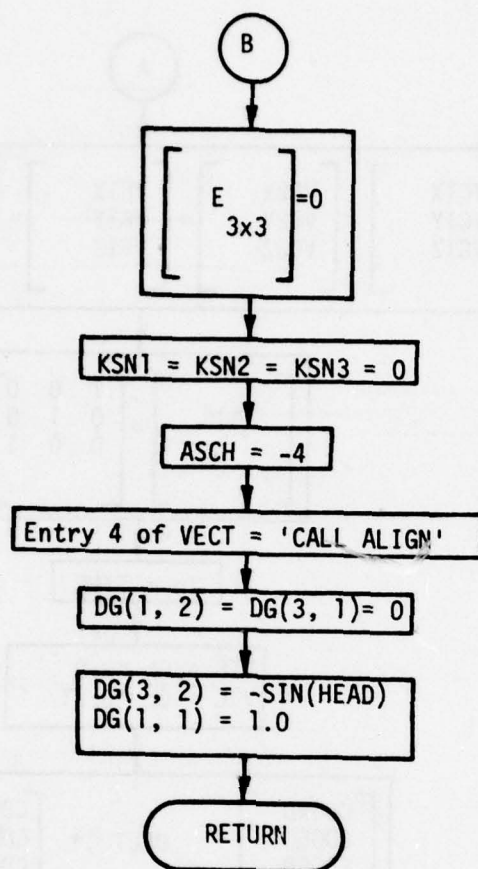
$$\begin{aligned} SGDL &= \sin(LAT) \\ CGDL &= \cos(LAT) \end{aligned}$$

$$\begin{aligned} AK1T &= CGDL * GL \\ AK2T &= SGDL * GL \end{aligned}$$

$$PHA = 0$$

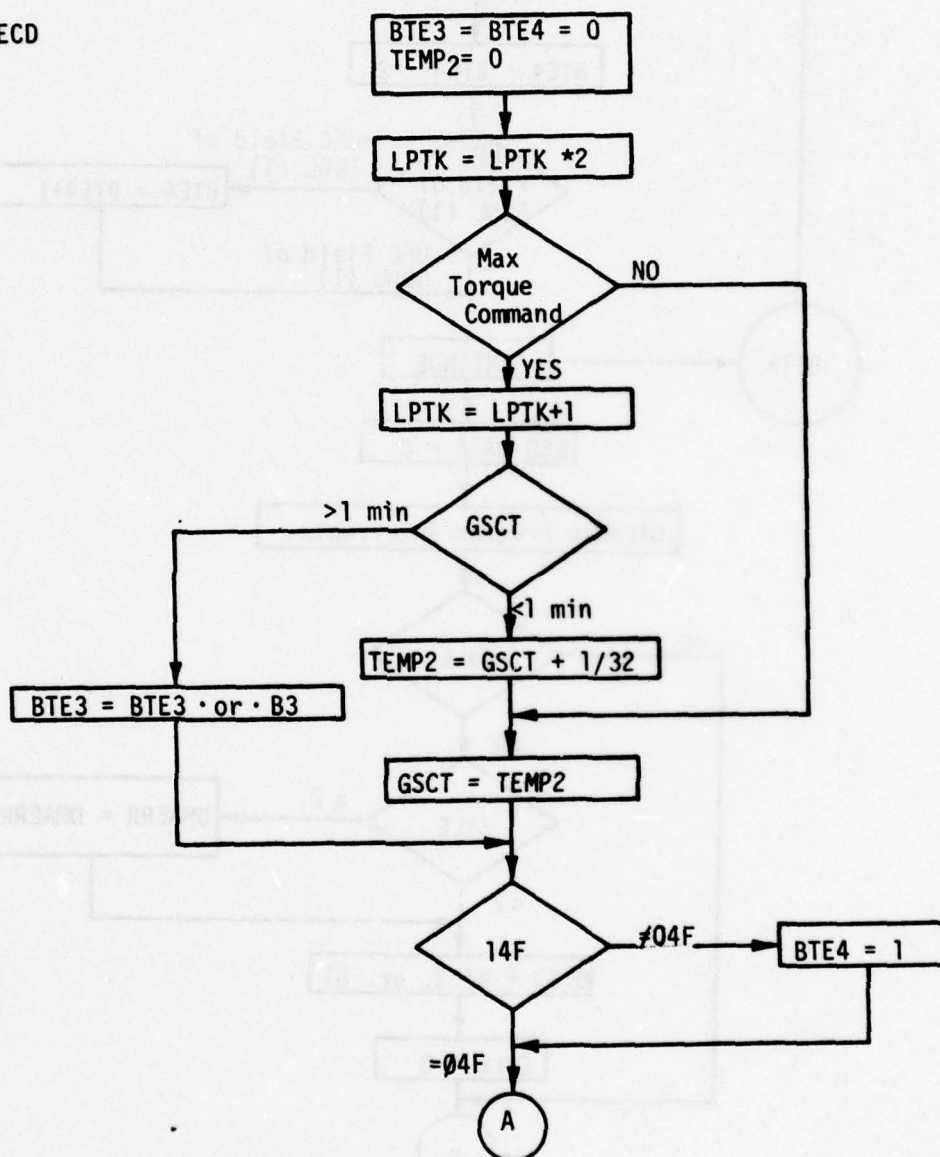
A



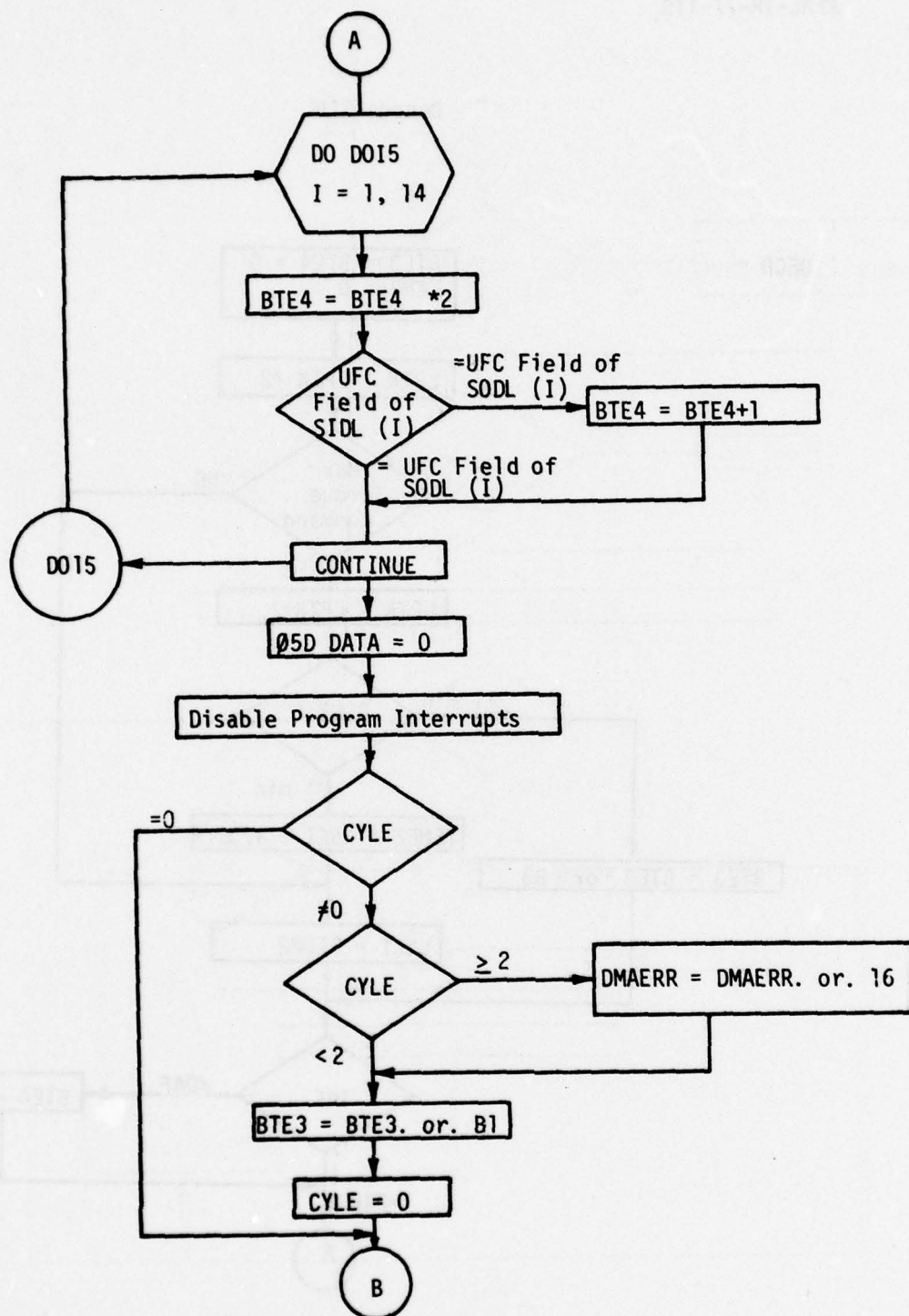


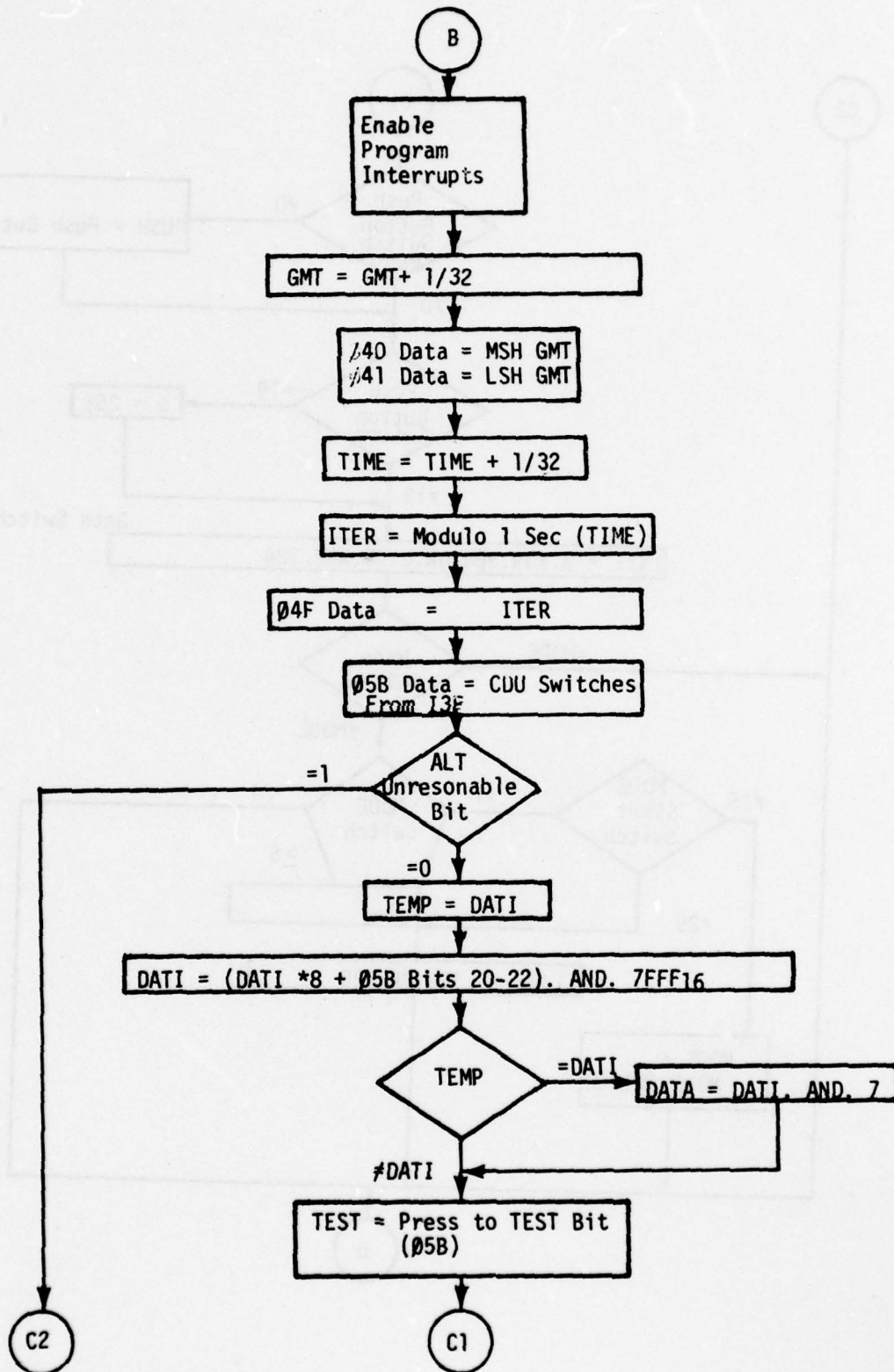
## Decode SIDL

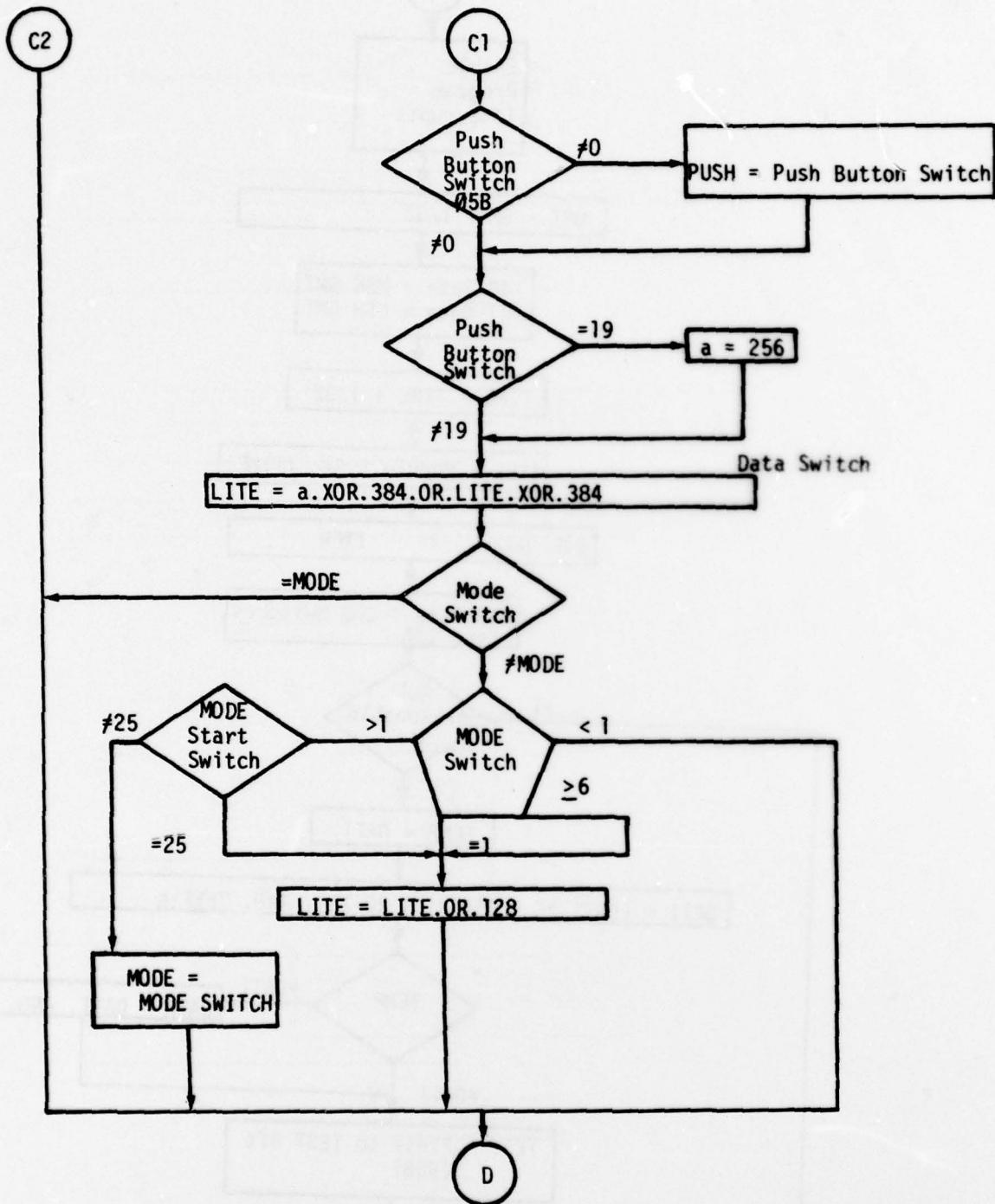
DECD

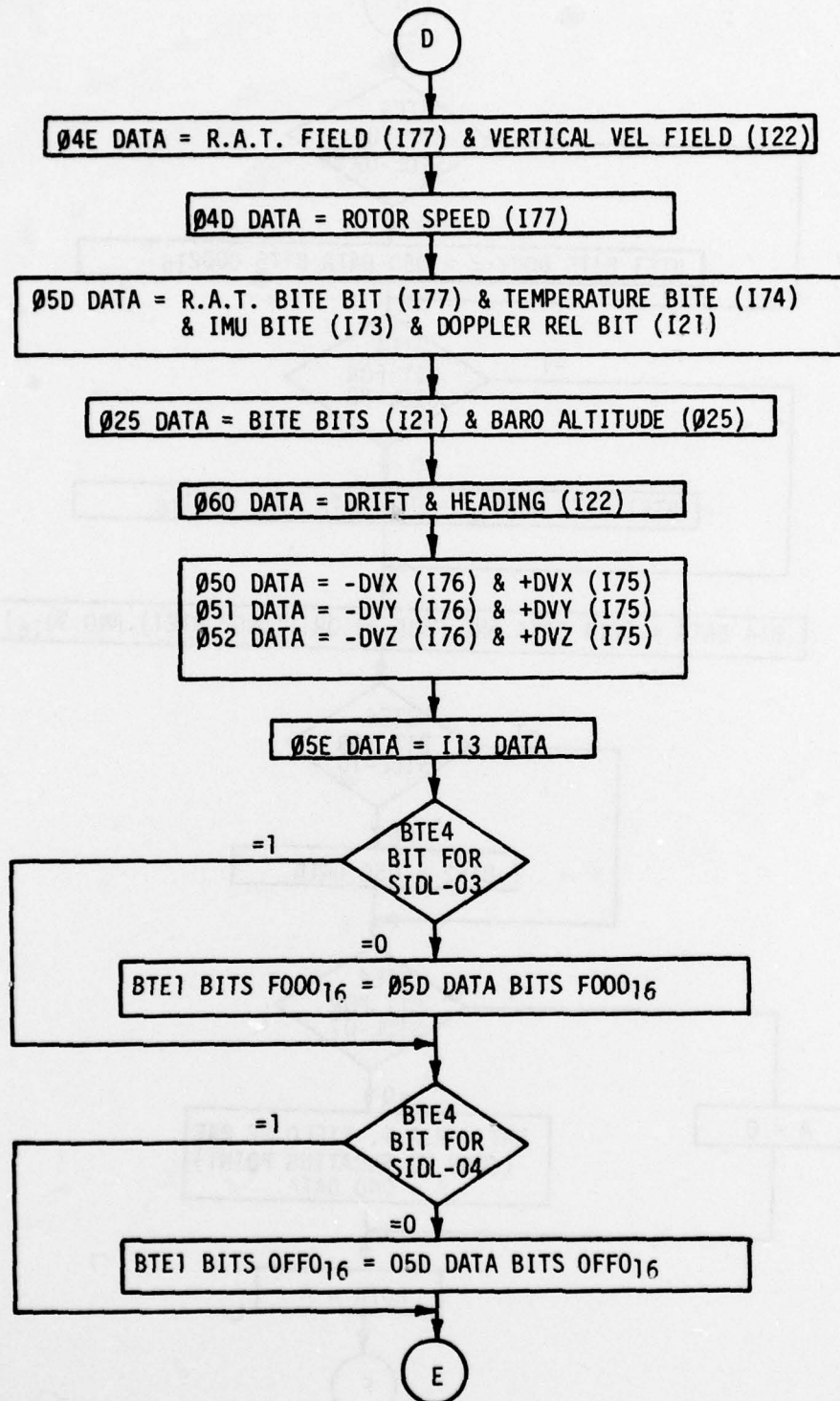




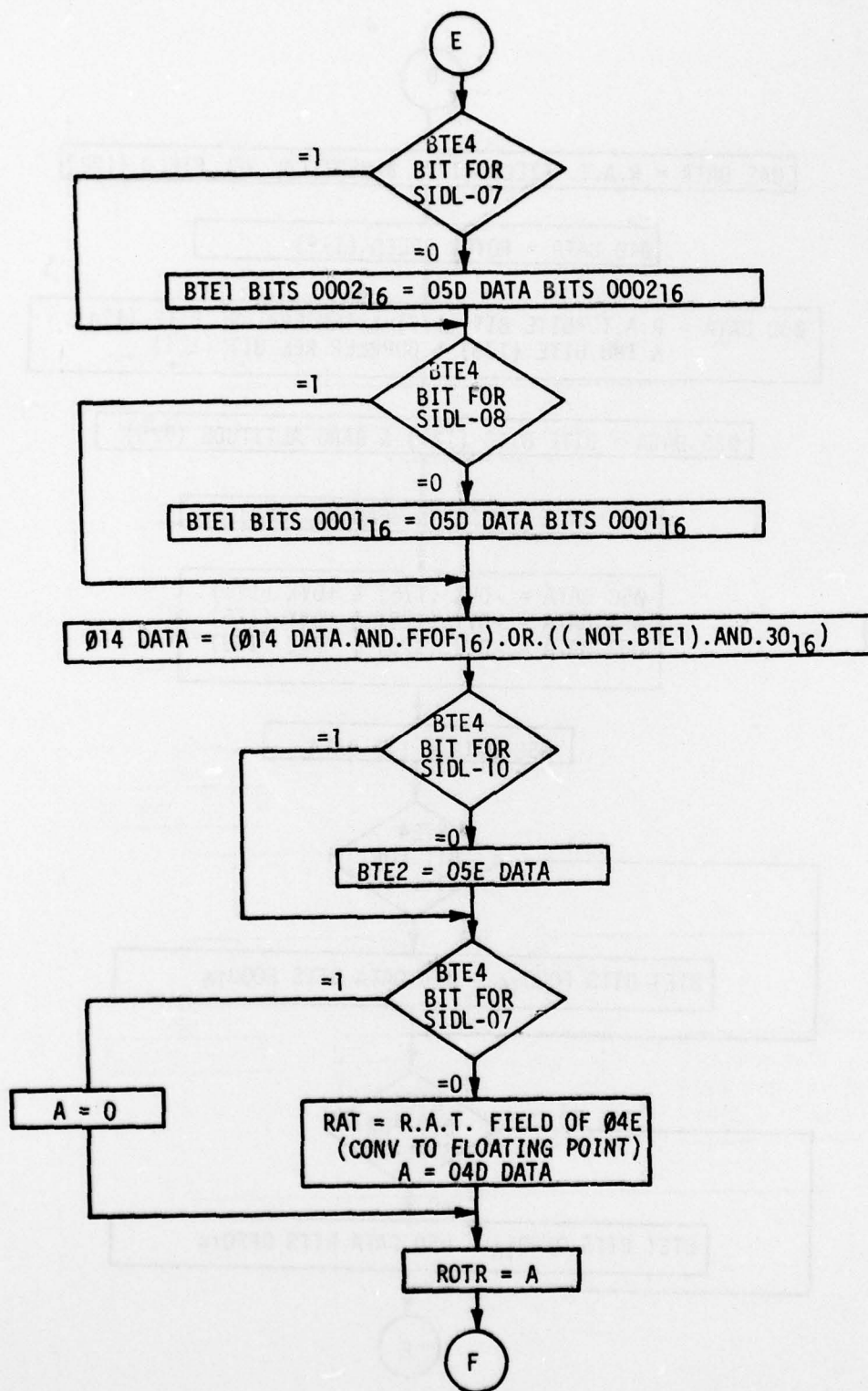


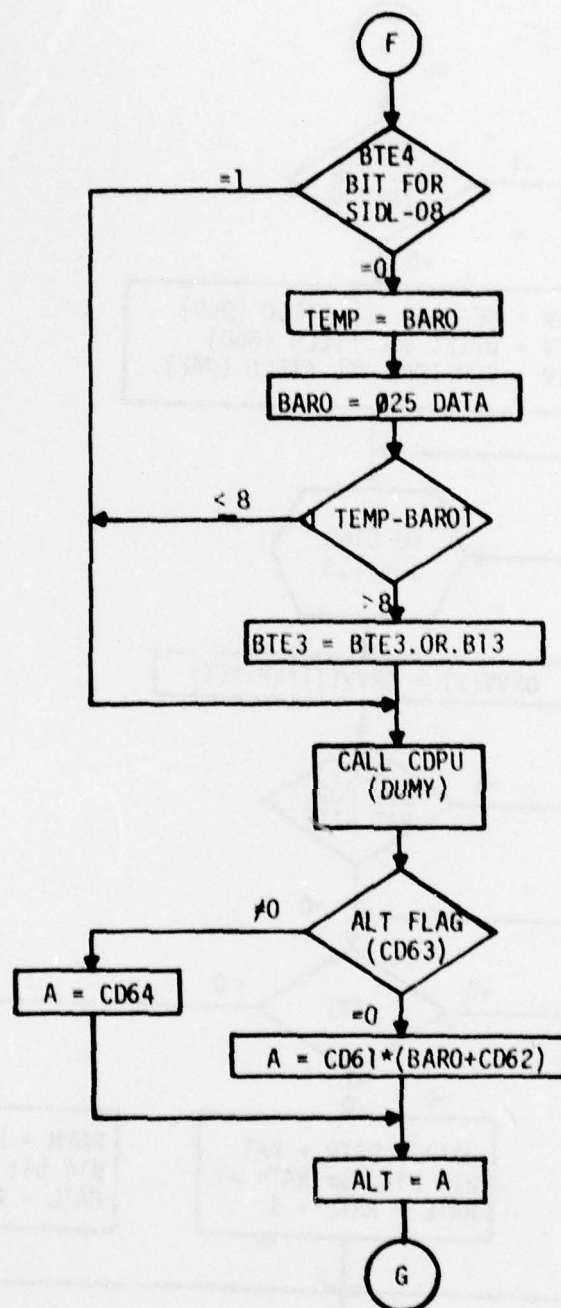


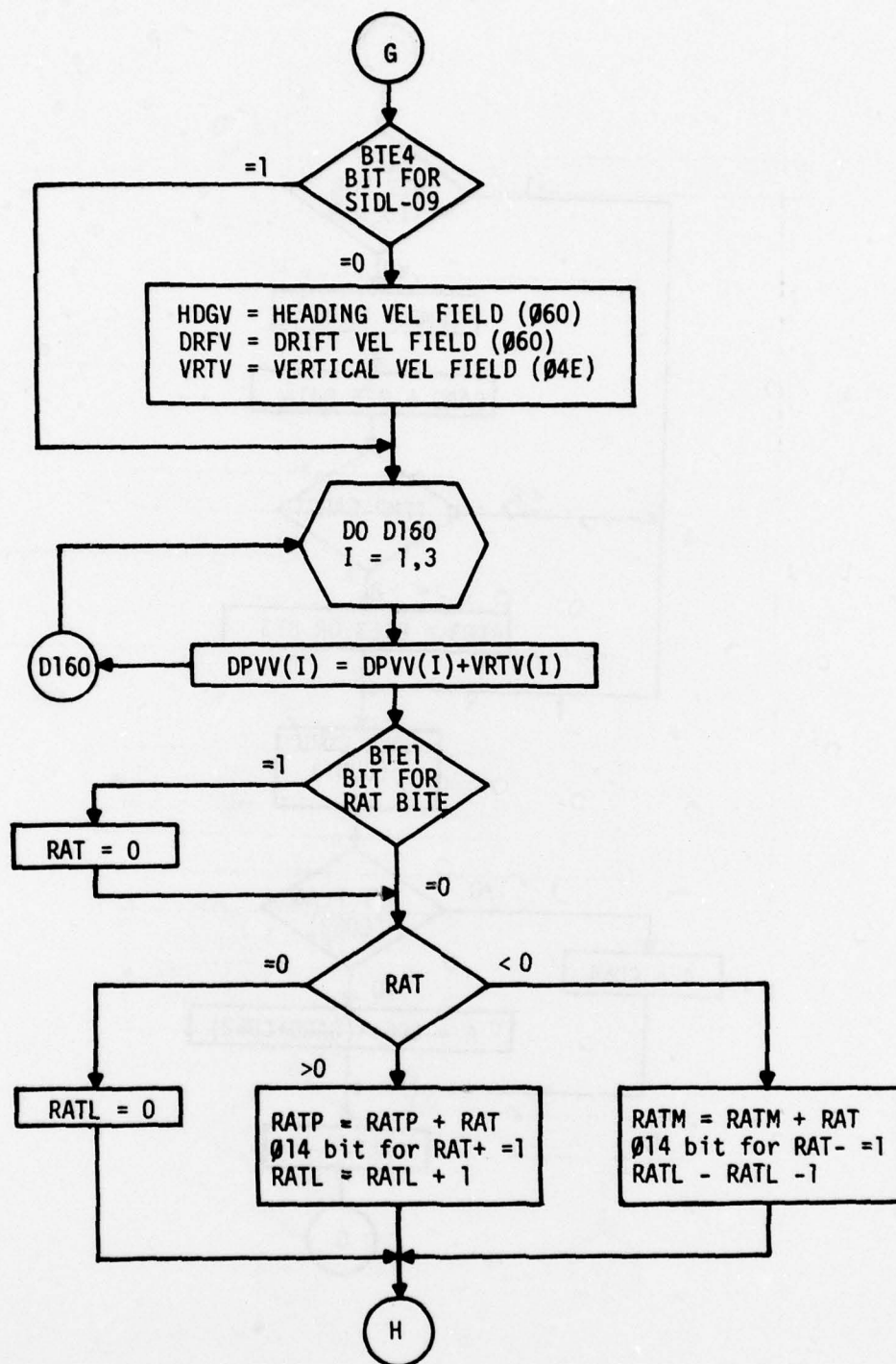


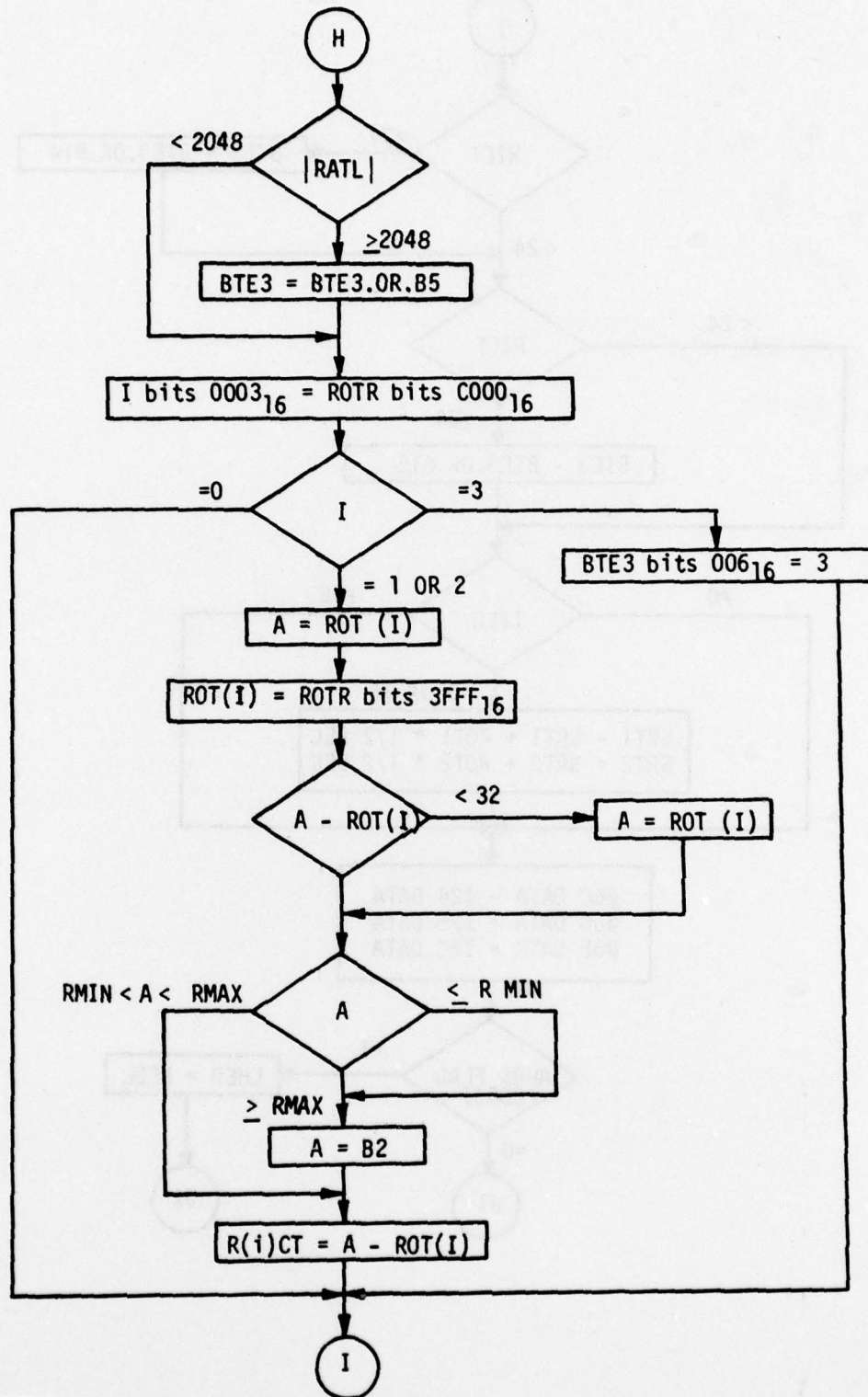




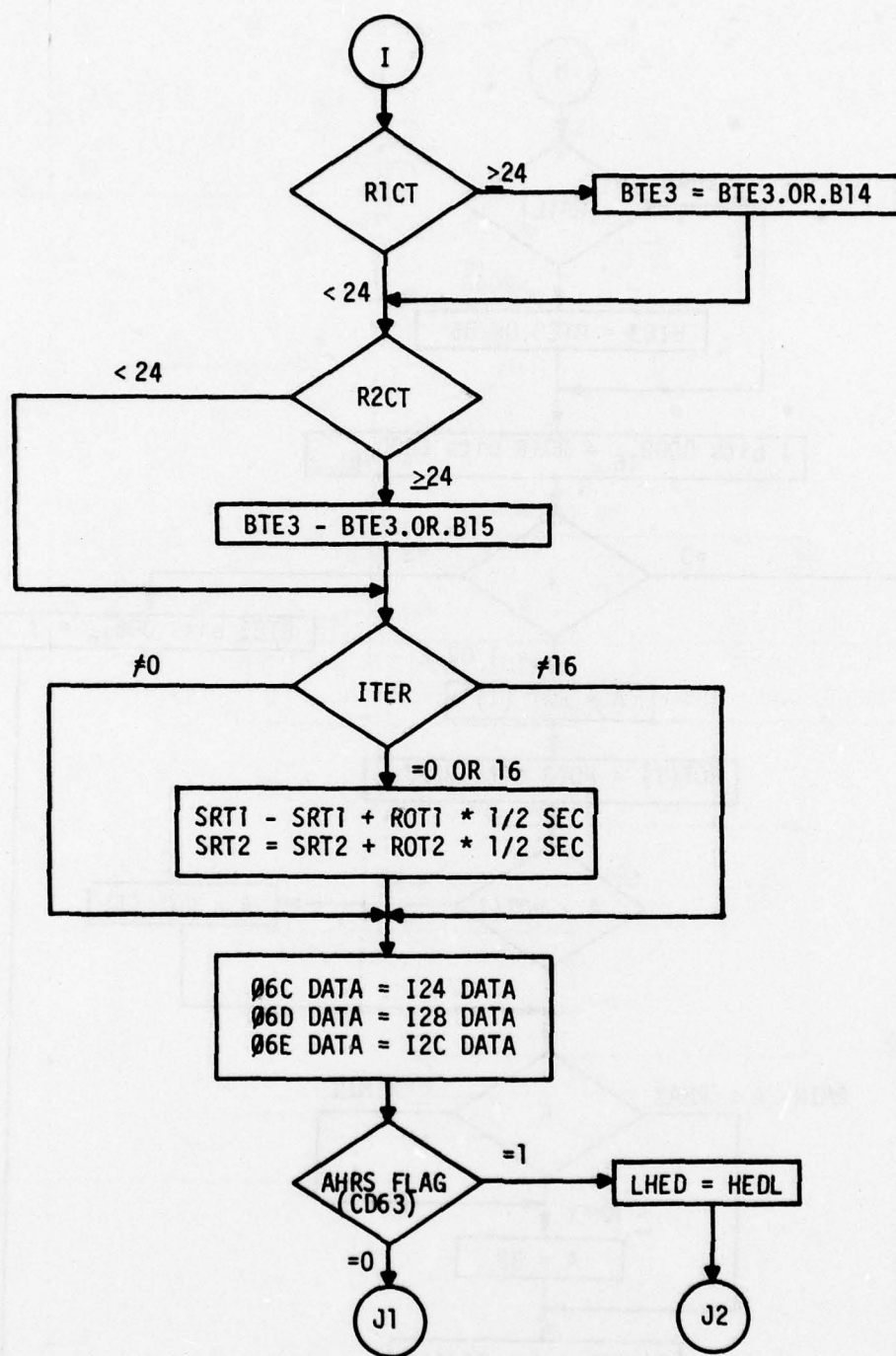


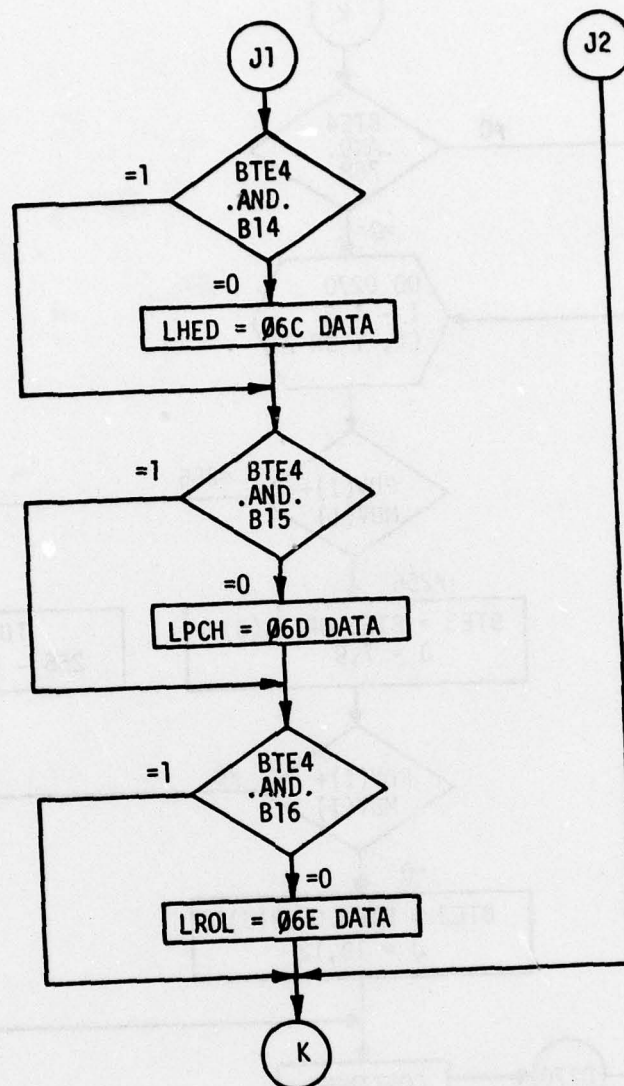


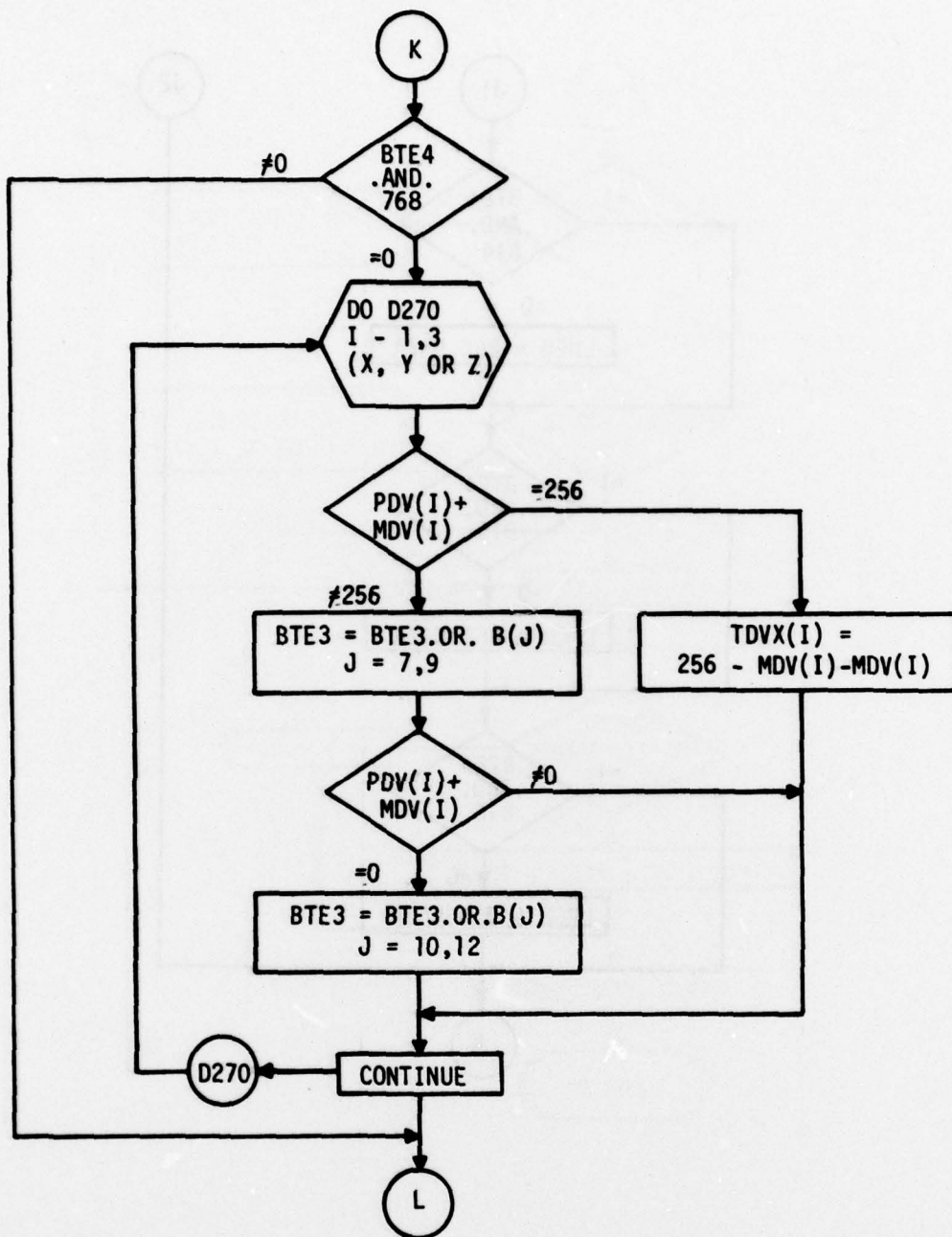


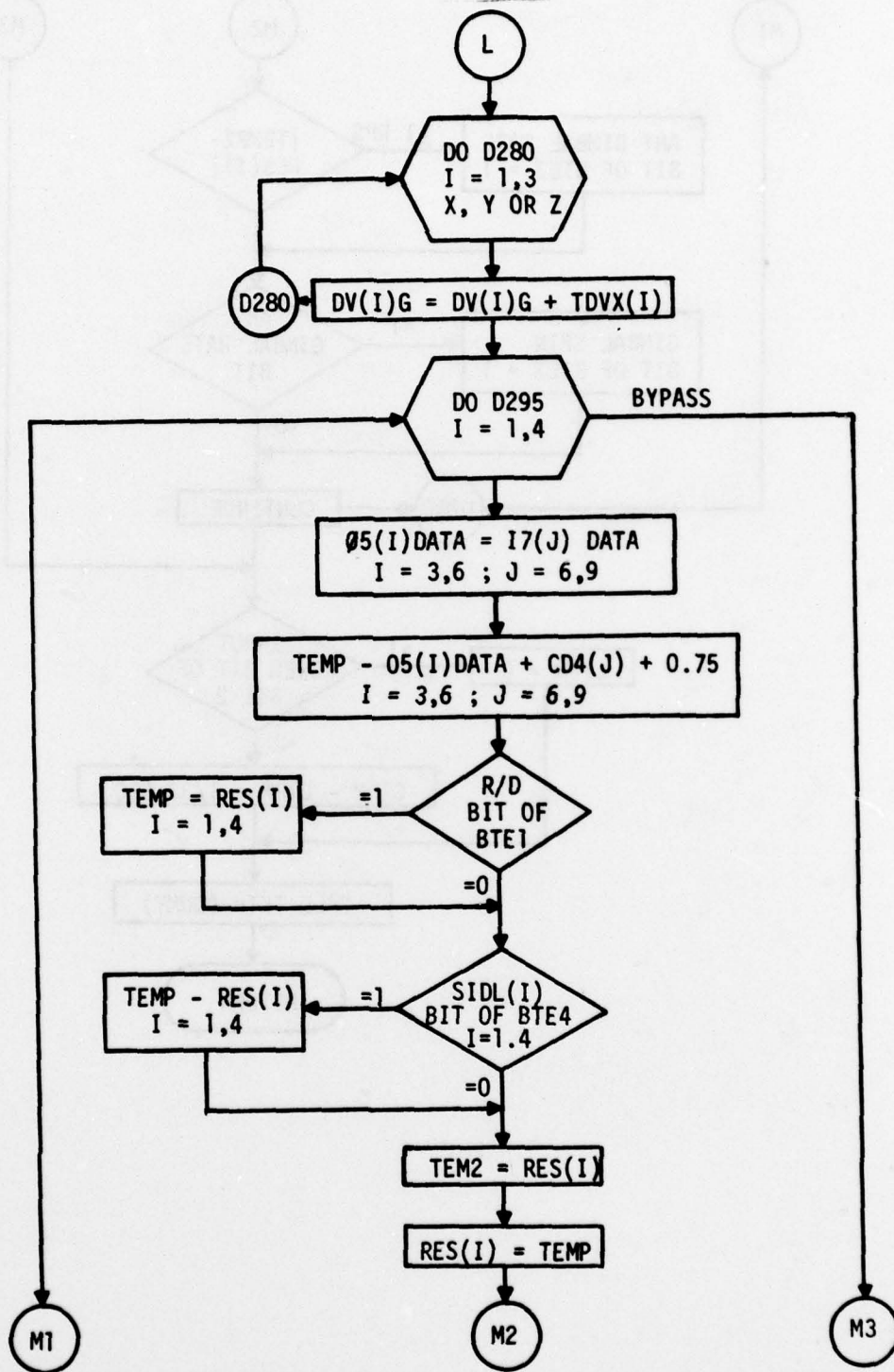




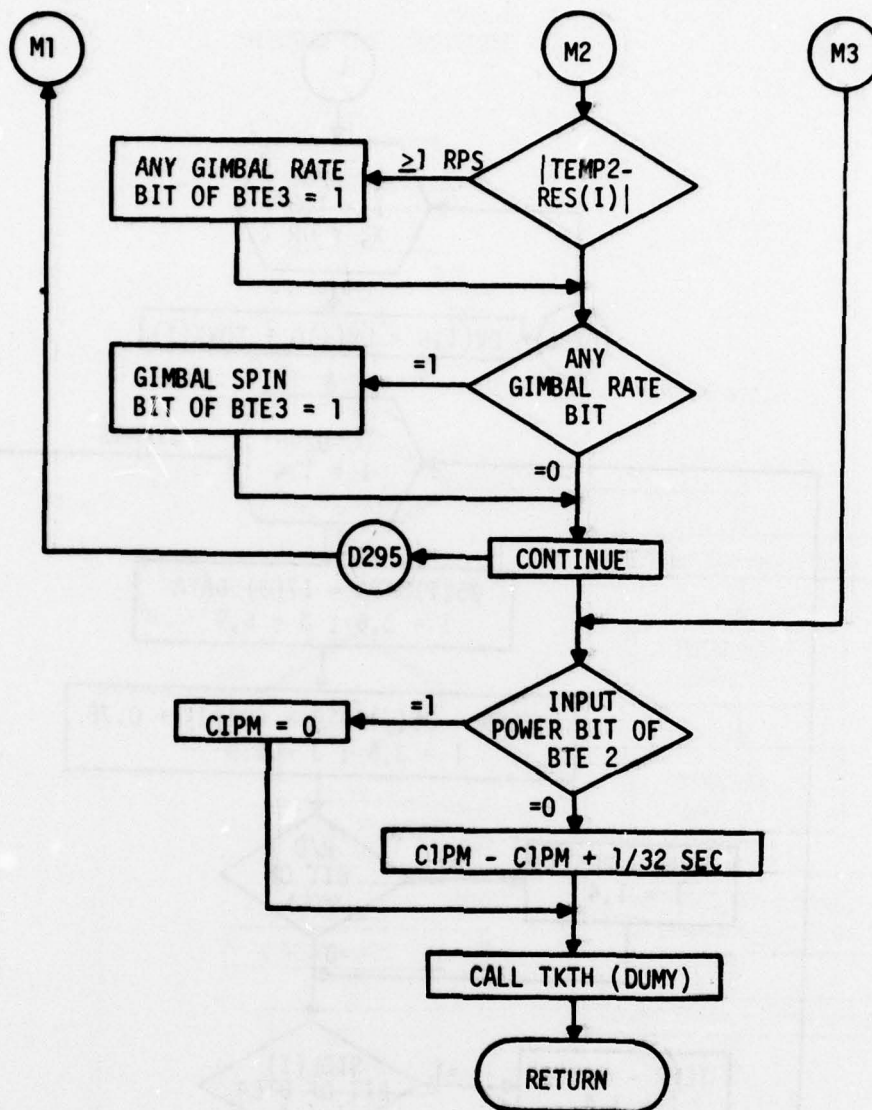




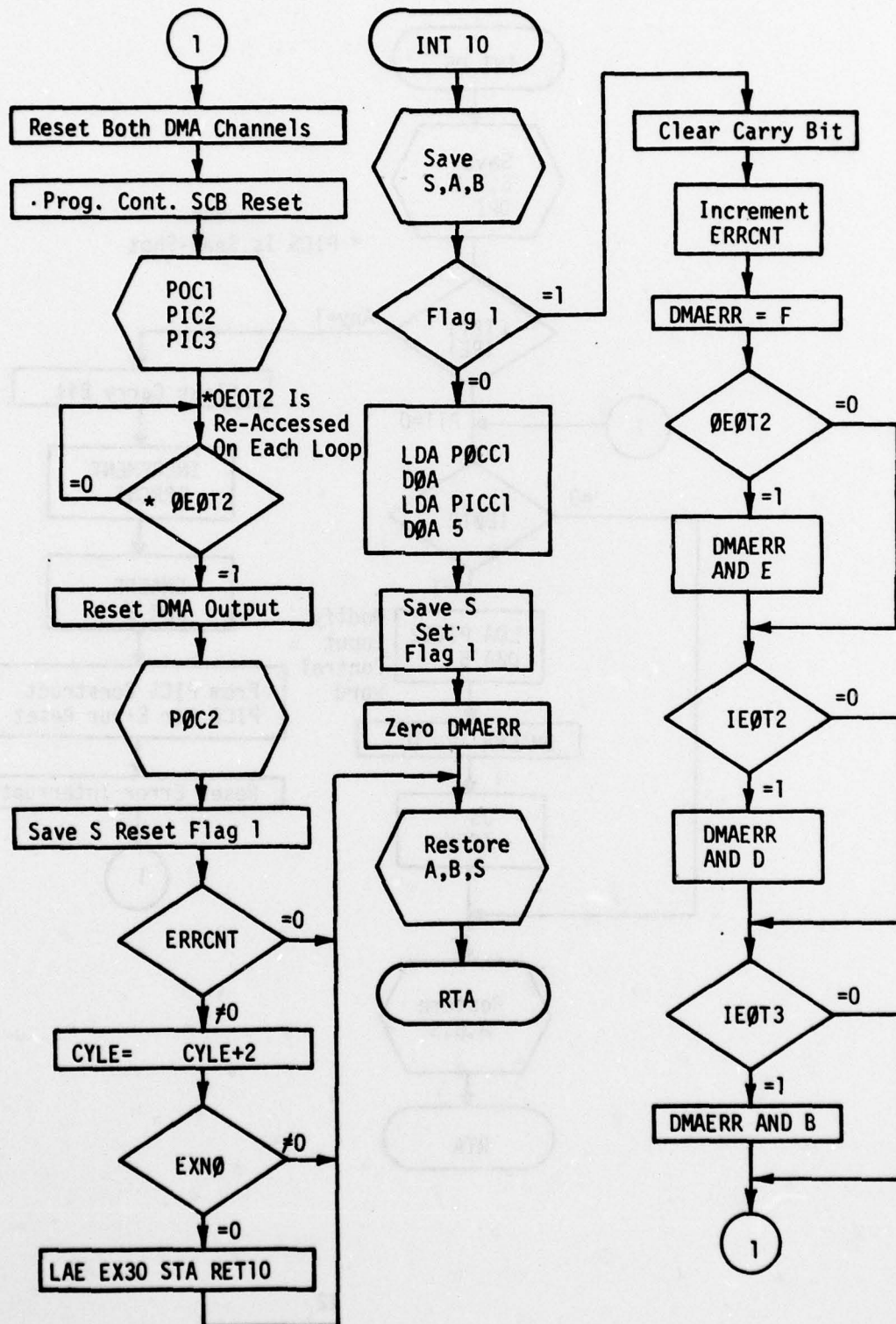




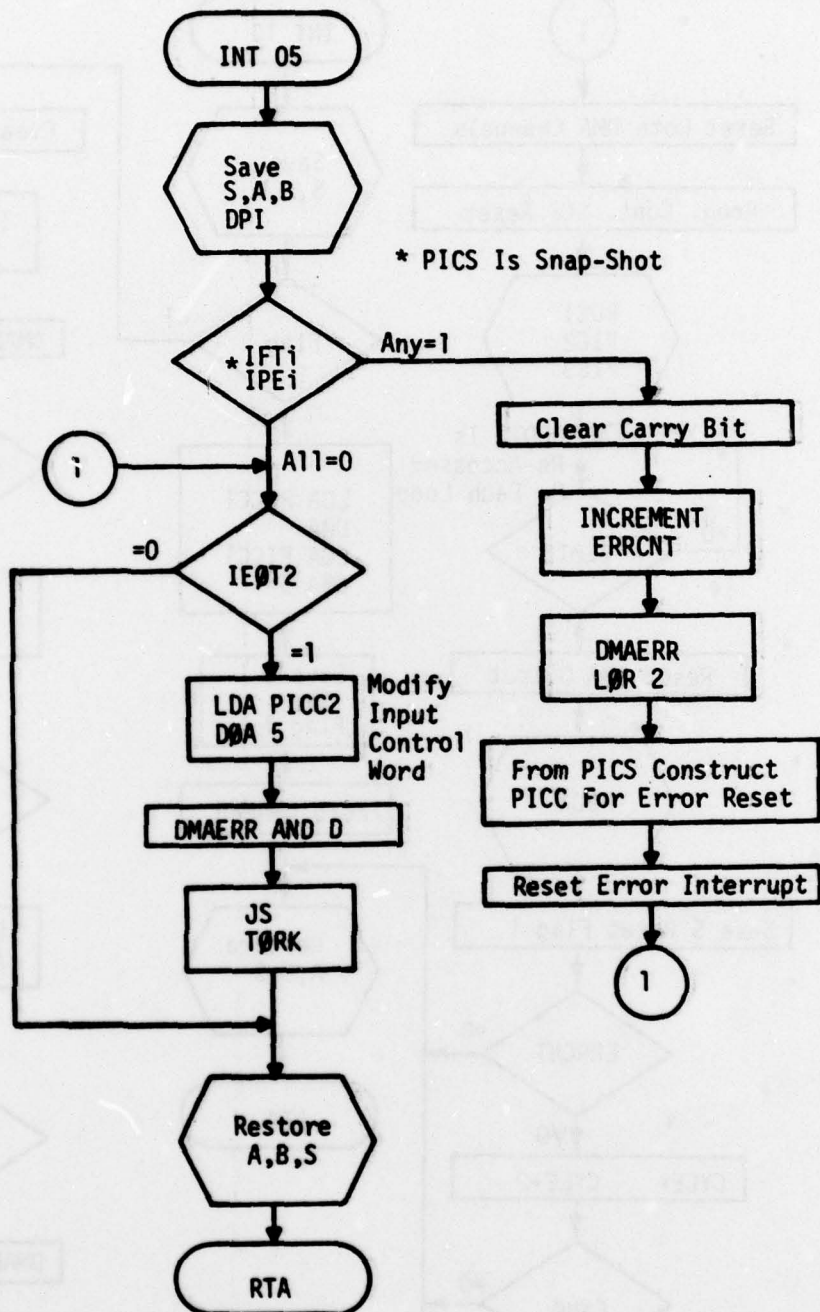




## INTERRUPT 10 ROUTINE [32 HZ]



## INTERRUPT 5 ROUTINE





## INTERRUPT 4 ROUTINE

